NASA CR 137,668

A KALMAN FILTER FOR THE STOLAND SYSTEM

by

Stanley F. Schmidt

April 1975

Distribution of this report is provided in the interest of information exchange. Responsibility for the contents resides in the author or organization that prepared it.

Prepared under Contract No. NAS2-8503 by

ANALYTICAL MECHANICS ASSOCIATES, INC.
80 West El Camino Real
Mountain View, California

for

AMES RESEARCH CENTER
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

ANALYTICAL MECHANICS ASSOCIATES, INC.

A KALMAN FILTER
for the
STOLAND SYSTEM

Report No. 75-10
Contract NAS2-8503
April 1975

Stanley F. Schmidt

Prepared for

National Aeronautics and Space Administration
Ames Research Center
Moffett Field, California 94035

ANALYTICAL MECHANICS ASSOCIATES, INC.
80 WEST EL CAMINO REAL
MOUNTAIN VIEW, CALIFORNIA 94040

SUMMARY

A simple Kalman filter for potential use in STOL navigation systems is described. The mathematical formulation of all the elements of the filter, its initialization and over all operation are presented. Simulation results show that on a typical approach flight to landing, the Kalman filter has much smaller errors during navigation on TACAN data and during transition from TACAN to MODILS data than a complementary filter.

Summary-type flow charts of the Kalman filter logic designed for the Sperry 1819A computer are presented. Also, the memory and real-time requirements of the Kalman filter and complementary filter are described. The Kalman filter is shown to gain its superior performance at the expense of real-time and memory required in the onboard computer. These requirements are not, however, excessive for modern airborne computers.

TABLE OF CONTENTS

ACKNOWLEDGMENTS

The baseline filter design mentioned in this report was developed by Mr. Rodney Wingrove of NASA Ames Research Center. Mr. Wingrove and Mr. Don Smith defined the guidelines and constraints of the overall effort.

Mr. Dick Sharp of the Sperry Flight Systems Software Support Group programmed the executive program described in the Appendix and gave substantial aid in the overall interface programming and checkout of the Kalman filter program. The author wishes to express appreciation to the above individuals for their contributions to the work described herein.

I. INTRODUCTION

This report describes the design and simulation results of a Kalman filter for potential application in the STOLAND. The STOLAND is a digital avionics system designed for testing terminal area guidance, navigation and control concepts for short-haul aircraft, and investigating operational procedures (see Ref. [1]). The current navigation system in STOLAND (see Ref. [2]) uses a complementary filter to blend ground navigation information with onboard inertial and airdata information. The Kalman filter described in this report operates in parallel with the complementary filter using the same onboard and navigation aid information, so that direct comparison of the performance of the two filters during simulation and flight tests is possible. The effort described herein carried the design through simulation validation; hence, comparative results of the two filters obtained from simulation are presented. As will be seen, the Kalman filter exhibits considerably smaller errors in the presence of biases in the ground navigation aid measurement or in the transition from one ground navigation aid to another. This improved performance is obtained at the expense of greater computer memory and real-time for the Kalman filter. Comparisons of the memory and real-time requirements of the two filter mechanizations is also presented.

The overall mechanization and design constraints of the Kalman filter are described in section III. Section IV gives the mathematical formulation of the design. Some preliminary results obtained from simulation are given in section V. A summary type description of the Kalman filter part of the 1819A software is given in the Appendix.

II. NOTATION AND SYMBOLS

The notation of "•" or "¨" over a symbol will have its customary meaning of differentiation with respect to time. The "^" (hat) mark over a symbol means the "estimated" or "computed" value of the symbolized quantity, while the "~" (tilde) indicates an error or small variation in the symbolized quantity. For instance, if x is the true value of the position component, then it may be written as the sum of the estimated component and the error.

$$x = \hat{x} + \tilde{x}$$

The notation $(\ )^T$ means the transpose of the quantity.

Roman Symbols

A	The transition matrix minus the identity matrix.
a_{xb}	The acceleration bias along the runway.
a_{yb}	The acceleration bias normal to the runway.
b_r	Bias in the TACAN range measurement.
b_ψ	Bias in the TACAN bearing measurement.
c	Coefficient used in data rejection.
dx	The error state with components defined in Table 3.1.
F_n	Jacobian matrix of the state rates with respect to random forcing functions.
F_x	Jacobian matrix of the state rates with respect to the estimated state.
H	Gradient of the measurement with respect to the estimated state.
H_m	H referred to the Kalman filter reference time $(H(t)\ \Phi\ (t;\ t_k))$

H_{ax}	H for along runway airspeed measurement.
H_{ay}	H for normal to runway airspeed measurement.
H_{ma}	H for MODILS azimuth measurement.
H_{mr}	H for MODILS range measurement.
H_{tb}	H for TACAN bearing measurement.
H_{tr}	H for TACAN range measurement.
K_m	Kalman filter gain.
n	Noise vector.
n_x	The number of residuals in a sum.
P	Covariance matrix.
q	Random error in a measurement.
q_{ax}	q for along runway airspeed measurement.
q_{ay}	q for normal to runway airspeed measurement.
q_{ma}	q for MODILS azimuth measurement.
q_{mr}	q for MODILS range measurement.
q_{tb}	q for TACAN bearing measurement.
q_{tr}	q for TACAN range measurement.
Q	Variance of the random error in a measurement.
Q_{ax}	Q for along runway airspeed measurement.
Q_{ay}	Q for normal to runway airspeed measurement.
Q_{ma}	Q for MODILS azimuth measurement.
Q_{mr}	Q for MODILS range measurement.
Q_{tb}	Q for TACAN bearing measurement
Q_{tr}	Q for TACAN range measurement.
r	Computed range from MODILS azimuth scanner.
r_1	Level component of distance from MODILS azimuth scanner.

t	Time
t_k	Reference time for the Kalman filter
$u(t_k)$	Discrete noise vector.
U	Square root covariance of noise vector, $u(t_k)$.
v_x	Along runway component of velocity.
v_y	Normal to runway component of velocity.
w_x	Along runway component of wind.
w_y	Normal to runway component of wind.
W	Square root covariance matrix, $WW^T = P$.
x	Along runway component of position.
y	Normal to runway component of position.
z	Down component of position.
x_m	Position of MODILS DME transponder and azimuth scanner with respect to the runway reference.
y_m	
z_m	
x_T	Position of TACAN station with respect to the runway reference.
y_T	
z_T	
$\ddot{x}_r$	Raw acceleration along runway.
$\ddot{y}_r$	Raw acceleration normal to runway.
y_m	Measurement residual.
Y	Mathematical model of a measurement.
Y_{ax}	Y for along runway airspeed measurement.
Y_{ay}	Y for normal to runway airspeed measurement.
Y_{ma}	Y for MODILS azimuth measurement.
Y_{mr}	Y for MODILS range measurement.

Y_{tb}	Y for TACAN bearing measurement.
Y_{tr}	Y for TACAN range measurement.

Greek Symbols

Δ	Time interval
Δ_k	Time interval between filter reference points.
Δy	Best estimate of residual, including past measurements.
Δy_m	The residual sum for an arbitrary measurement.
ζ	Vector defined in Potter's algorithm [4.14].
η	Vector defined in Potter's algorithm [4.14].
κ	Scalar defined in Potter's algorithm [4.14].
σ	Standard deviation (std) of a random variable.
σ_a	Std of noise in acceleration bias model.
σ_m	Std of a residual sum.
σ_r	Std of noise in TACAN range bias model.
σ_v	Std of acceleration noise for model compensation.
σ_w	Std of noise in wind model.
σ_y	Std of measurement residual.
σ_ψ	Std of noise in TACAN bearing bias model.
τ_a	Time constant in acceleration bias model.
τ_r	Time constant in TACAN range bias model.
τ_w	Time constant in wind model
τ_ψ	Time constant in TACAN bearing bias model.
Φ	The state transition matrix.
Φ_u	Matrix of state sensitivity to forcing functions.

Ψ_a Magnetic heading of the aircraft.

Ψ_r Magnetic azimuth of the runway.

∇_x Gradient of function with respect to state.

III. OVERALL MECHANIZATION AND DESIGN CONSTRAINTS

3.1 Overview

Prior to initiation of the work described in this report, NASA had conducted an in-house study effort to assess the potential performance improvements in using a Kalman filter in the STOLAND system. This in-house study had configured the overall mechanization with respect to factors such as:

(1) the state variables to be included in the filter design;

(2) the measurements to be processed by the filter, and

(3) the operating regions in the flight-envelope where the Kalman filter was to operate.

In addition, the filter was to be designed to operate within the available memory and real-time in the Sperry 1819A computer without effecting the operation of any of the existing software. This latter constraint meant that the onboard filter had to be designed to operate with less than 4000 words of memory and less than about 25% real-time of the 1819A computer.

Another constraint was that the Kalman filter should use the same measurement data used in the complementary filter. This constraint allows a reasonable comparison of the performance of the Kalman filter with the complementary filter. Improved performance with the more sophisticated filter is to be expected at the expense of more memory and real-time utilization. By using the same measurement data in both filters, a reasonable trade-off between performance and complexity was anticipated.

This section presents the overall mechanization of the NASA baseline design and the error state vector developed by this design.

3.2 Overall Mechanization

Figure 3.1 presents the NASA baseline design for the onboard Kalman filter. The external sensor measurements used by the filter include:

(1) attitude and heading,
(2) body-mounted accelerometers,
(3) air data,
(4) TACAN range and bearing, and
(5) MODILS range and azimuth.

Existing software in the 1819A computer calculates the acceleration in a runway reference which is fed to the navigation equations, and the true airspeed in runway reference which is input to the Kalman filter algorithm. The TACAN range, bearing and their validity flags and the MODILS range, azimuth and their validity flags are also formatted and scaled by the existing software before being sent to the Kalman filter.

The navigation equations appropriately integrate the runway referenced acceleration at 10Hz to form the position and velocity estimates. These quantities as well as the other state variables included in the estimated state are also input to the Kalman filter algorithm.

The Kalman filter algorithm computes the error in the estimated state from the input information which in turn is fed back and added to the state estimate carried by the navigation equations. The Kalman filter algorithm requires lengthy calculations, so in order to keep its operations

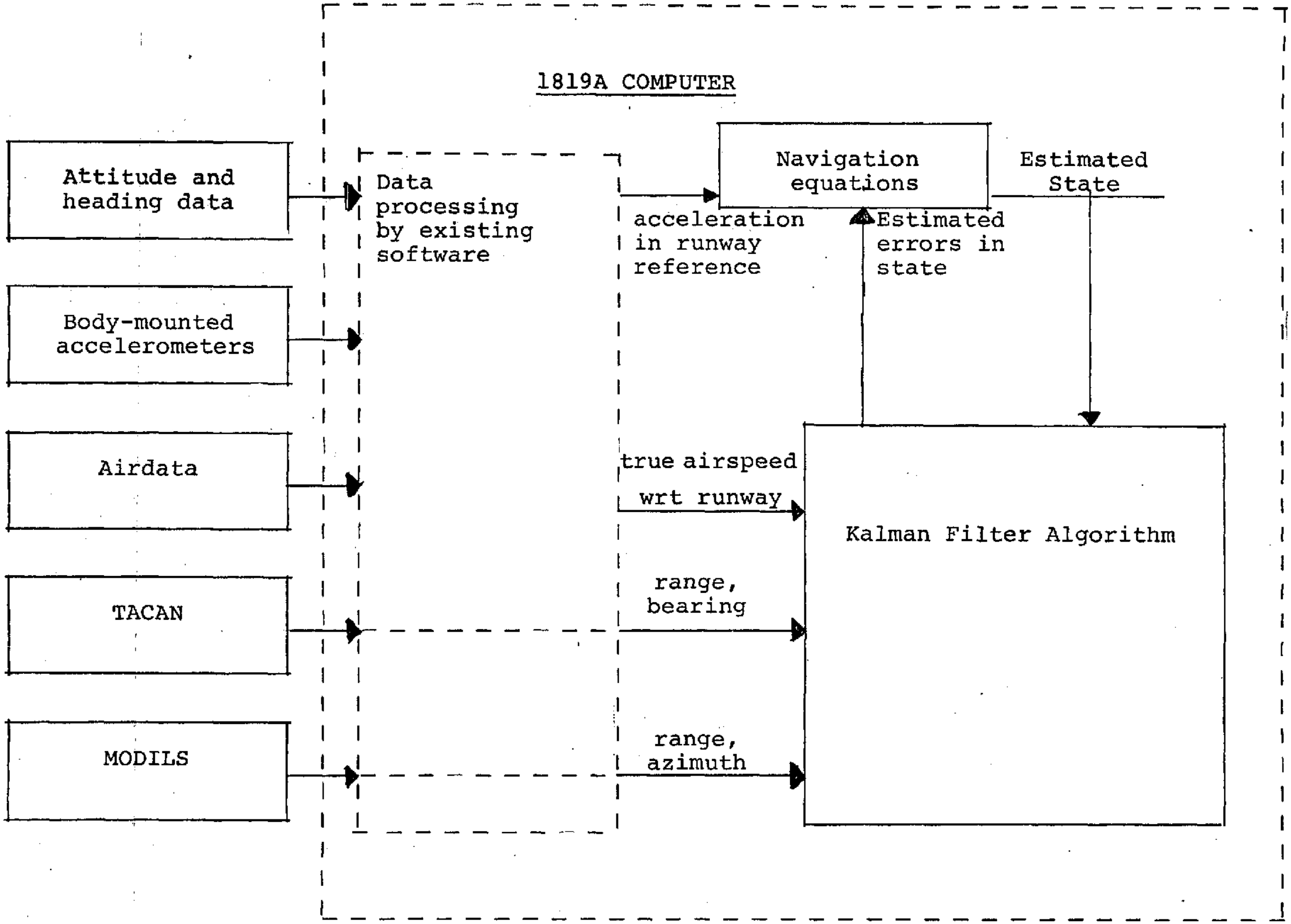

1819A COMPUTER
Attitude and heading data
Body-mounted accelerometers
Airdata
TACAN
MODILS
Data processing by existing software
acceleration in runway reference
Navigation equations
Estimated State
Estimated errors in state
true airspeed wrt runway
Kalman Filter Algorithm
range, bearing
range, azimuth

Figure 3.1. NASA Baseline Design of Overall Mechanization

within the available machine time, the estimated error state is calculated at a 1Hz frequency.

In order to use all the navigation and data which is available at 10Hz, the Kalman filter contains preprocessing routines for "averaging" the navaid data. These operations will be described in section IV.

3.3 Error-State Vector

The error-state vector used in the Kalman filter of Figure 3.1 is given in Table 3.1.

Table 3.1. Error-State Vector

Element	Symbol	Definition
1	$\tilde{x}$	position error along the runway
2	$\tilde{y}$	position error normal to runway
3	$\tilde{v}_x$	velocity error along the runway
4	$\tilde{v}_y$	velocity error normal to runway
5	$\tilde{a}_{xb}$	bias error in acceleration along runway
6	$\tilde{a}_{yb}$	bias error in acceleration normal to runway
7	$\tilde{b}_r$	bias error in TACAN range measurement
8	$\tilde{b}_\psi$	bias error in TACAN bearing measurement
9	$\tilde{w}_x$	error in estimate of wind along the runway
10	$\tilde{w}_y$	error in estimate of wind normal to runway

As may be noted, the state variables estimated by the filter only involve the level (x, y) components of navigation quantities. A navigation system requires a vertical channel also; however, the NASA baseline design had relegated studies associated with a 3-axis system to a later effort contingent on the results of the simpler level-axis design.

The first four elements of the error-state are level components of position and velocity. The fifth and sixth components are acceleration biases which are treated as exponentially correlated noise variables. These two biases account in part for the errors in the simple strapped-down inertial system. These acceleration biases are also calculated by the complementary filter (see Ref. 2).

The seventh and eighth error-state variables are biases in the range and bearing data from the TACAN station. These quantities are also treated as exponentially correlated noise variables.

The ninth and tenth error-state variables are errors in the two components of winds. These are treated as exponentially correlated noise variables to account for wind changes during the approach flight profile. The airdata was used in the Kalman filter to both provide the estimates of winds and for the potential help in smoothing the ground velocity estimates obtained from the filter.

IV. FILTER IMPLEMENTATION

The navigation equations (see Figure 3.1) integrate the accelerometer data at a high rate (10 Hz) to keep the estimate of state current. Measurements of state from the TACAN and MODILS receivers, and the computed true air speed feed the Kalman filter of Figure 3.1. The filter uses the measurements and the current estimate of state to calculate an incremental change in the current estimate which is in turn added to the current estimate.

This section starts by summarizing the basic filter equations. Next, the navigation equations used for keeping the estimate of state current and the error equations for use in the time update of the filter are summarized. Next, the formulations are developed for processing measurements by the filter. The section concludes with a brief discussion of how the system is initialized.

4.1 Summary of the Kalman Filter Algorithm

The Kalman filter algorithm involves a very large number of machine operations. The basic algorithm separates into the following:

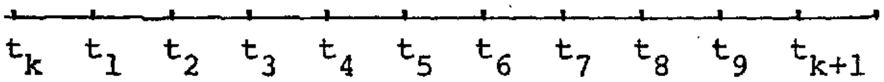
1. Calculations associated with a change in the time reference of the filter, and
2. Calculations of the incremental state estimate from the measurement information.

In order to keep the number of operations within the capabilities of current airborne computers the time reference is changed at a moderately low frequency (e.g., 1Hz) and average type measurements are processed at a moderately low frequency (e.g., 1 Hz).

It is a well know fact that a practical filter design must minimize effects caused by: (1) numerical calculation errors such as truncation, and (2) modeling errors resulting from various approximations.

Past experience has shown that the square root implementation of the optimal filter algorithm (references [3] and [4]) can reduce the effects of numerical errors to insignificant levels. The square root implementation is therefore incorporated in the design. Modeling errors are compensated by the appropriate use of random forcing functions. This technique causes the more recent measurements to be weighted more than past measurements; therefore, the estimate tends to follow the more recent measurements.

The onboard filter operates in a manner illustrated in the sketch below:



At the start of the sequence we assume that the filter has its covariance matrix referenced to time t_k. At the times t_k, t_1......t_9 measurements are accepted and residual sums and partials are computed and saved in the preprocessing routines. After accepting the measurement at time t_9 the residual sums and partials are transferred to arrays for processing by the filter and the locations used for the preprocessing are cleared for use in preprocessing the measurements at t_{k+1} and the subsequent measurements. The residual sums are processed by the filter and the incremental state change computed in lower priority logic. When these lower-priority calculations have been completed, the logic sets markers which cause the higher priority logic to pick up the state change and add it to the system. Meanwhile, the lower priority logic updates the covariance matrix to the time t_{k+1} and readies the filter for processing the residual sums in the next interval. The onboard program operations for executing this logic in the Sperry 1819A computer are described in the Appendix.

Time Update Operations

The error state given in Table 3.1 is assumed to obey the vector differential equation,

$$d\dot{x} = F_x dx + F_n n, \tag{4.1}$$

where

dx = the n(10) component error state vector

F_x = an nxn matrix

F_n = an nxm matrix

n = an m vector of random forcing functions for compensation of error growth caused by unmodeled error sources.

An approximate solution of (4.1) is,

$$dx(t_{k+1}) = \Phi(t_{k+1};t_k)dx(t_k)+\Phi_u(t_{k+1};t_k)u(t_k) \tag{4.2}$$

where

Φ = the transition matrix,

Φ_u = the forcing function sensitivity matrix,

$u(t_k)$ = a constant (in the interval t_k to t_{k+1}) vector for approximating the effects of the noise vector, n, of (4.1).

The covariance matrix for the error state at time t_k is given by,

$$P(t_k) = W(t_k)W(t_k)^T = E(dx(t_k)dx(t_k)^T), \tag{4.3}$$

where

$W(t_k)$ = the square root covariance. (W^T is calculated in the square root implementation of the filter),

$E()$ = the expected value operator.

We assume that $u(t_k)$ is a random independent vector such that

$$\begin{aligned} E(u(t_{k+i})u(t_{k+\ell})^T) &= 0 & i \neq \ell \\ &= U(t_{k+i})U(t_{k+i})^T & i = \ell \end{aligned} \tag{4.4}$$

The appropriate use of the expected value operator with Equation (4.2) gives the time update of the covariance matrix,

$$P(t_{k+1}) = W(t_{k+1})W(t_{k+1})^T = \begin{bmatrix} \Phi W(t_k) & \Phi_u U \end{bmatrix} \begin{bmatrix} W(t_k)^T \Phi^T \\ U^T \Phi_u{}^T \end{bmatrix} \tag{4.5}$$

We see from Equation (4.5) that we can form $W(t_{k+1})^T$ in two steps as follows:

$$W(t_{k+1})^T = \begin{bmatrix} W(t_k)^T \Phi(t_{k+1};t_k)^T \\ \\ U(t_k)^T \Phi_u(t_{k+1};t_k)^T \end{bmatrix} \tag{4.6}$$

The matrix, $W(t_{k+1})^T$ of Equation (4.6) has dimension (n+m) x n. The Householder algorithm described in References [3] and [4] can be used to reduce this matrix to an upper triangular form; that is, all the terms below the diagonal are zero in the reduced matrix. The matrix reduction algorithm leaves the product WW^T invariant.

Section 4.2 of this report gives details on the matrices Φ, Φ_u, and U involved in the time update operations.

Measurement Processing Operations

The measurement residual, y_m, is defined by,

$$y_m(t) = \text{measured value} - Y(\hat{x},t), \tag{4.7}$$

where

$Y(\hat{x},t)$ = the computed value of the measurement based on the current estimate of state indicated by the output of the navigation equations.

It is assumed that the residual is related to the error state by,

$$y_m(t) = Hdx(t) + q, \tag{4.8}$$

where

q = the random error in the measurement.

Since we may have an estimate of dx given by $d\hat{x}$ from other measurements, we write (4.8) as,

$$\Delta y = y_m(t) - Hd\hat{x}(t) = Hd\tilde{x}(t) + q \tag{4.9}$$

= our best estimate of the residual including any past measurements whose effect is not yet included in the estimate of state carried by the navigation equations.

As was mentioned in the beginning of this section, the filter operates with the reference time t_k for measurements in the interval t_k to t_{k+1}. This means that we estimate $d\hat{x}(t_k)$ rather than $d\hat{x}(t)$. Hence we make the assumption for $t_k<t<t_{k+1}$ that

$$d\tilde{x}(t) = \Phi(t;t_k)d\tilde{x}(t_k). \qquad (4.10)$$

Equation (4.9) can then be written as,

$$\Delta y = y_m(t) - H\Phi(t;t_k)d\hat{x}(t_k). \qquad (4.11)$$

One may note that (4.10) and (4.2) are in disagreement since the influence of the random forcing function, $u(t_k)$, is ignored in (4.10). The effect of this simplification is that measurements are not quite optimally weighted. Experience has shown that this simplification is justified if the batch interval $\Delta_k = t_{k+1}-t_k$ is short compared to natural periods of error growth in the navigation equations.

We let,

$$H_m = H\Phi(t;t_k),$$

$$K_m = W(t_k)W(t_k)^T H_m^T/(H_m W(t_k)W(t_k)^T H_m^T+Q), \qquad (4.12)$$

$$Q = E(q^2).$$

Then the estimate of the error state following inclusion of the measurement is,

$$d\hat{x}(t_k)_a = d\hat{x}(t_k)_b + K_m\Delta\, y. \qquad (4.13)$$

In (4.13) the subscript notation $(\ )_b$ and $(\ )_a$ means before and after inclusion of the measurement, respectively.

The square root covariance matrix, $W(t_k)^T$, after inclusion of the measurement is given by,

$$W(t_k)_a^T = W(t_k)_b^T - \zeta\eta^T/\kappa, \qquad (4.14)$$

where

$$\zeta = W(t_k)_b^T H_m^T,$$

$$\eta = W(t_k)_b \zeta,$$

$$\kappa = (\zeta^T\zeta+Q)(1+\sqrt{Q/(\zeta^T\zeta+Q)}\ ).$$

Equation (4.14) is referred to as Potter's algorithm (see Reference [3]).

In order to minimize the number of measurement operations we will use accumulated residuals over a one-second period. This means that the residual, $y_m(t)$, of (4.7) is the accumulated residual for all the measurements of the same type in a one-second interval. The partial (H_m of (4.12)) is calculated and accumulated simultaneously with the residual accumulation. The Q of Equation (4.12) is the assumed variance of the random error in the accumulated residual (residual sum). More details on the residual sum processing are given in section 4.4.

4.2 Navigation and Error Equations

The navigation equations used to keep the state estimate current involve a double integration of

$$\begin{pmatrix} \hat{\ddot{x}} \\ \hat{\ddot{y}} \end{pmatrix} = \begin{pmatrix} \ddot{x}_r \\ \ddot{y}_r \end{pmatrix} + \begin{pmatrix} \hat{a}_{xb} \\ \hat{a}_{yb} \end{pmatrix} \qquad (4.15)$$

where, $\begin{pmatrix} \ddot{x}_r \\ \ddot{y}_r \end{pmatrix} =$ raw acceleration in runway reference computed by existing software (see Ref. [2]).

$$\begin{pmatrix} \hat{a}_{xb} \\ \hat{a}_{yb} \end{pmatrix} = \text{estimates of acceleration bias.}$$

The first integration of (4.15) gives the estimated velocity with respect to the runway and the second integration gives the estimated position with respect to the runway. Equation (4.15) is an approximation which is valid for a "flat" non-rotating earth. The errors resulting from this approximation are negligible in comparison to the errors caused by inertial hardware components (that is, the errors in the attitude and heading references and the errors in the body-mounted accelerometers). In the onboard program the raw acceleration data is accepted at 20Hz and averaged. The average acceleration is integrated at 10Hz.

Error Equations

The vector form of the error equations was given in (4.2) where the error state vector, dx, was defined in Table 3.1. In the subsequent summary we assume that elements of the noise vector $u(t_k)$ are all independent variables with unit variance. The gain associated with the noise is included in the constants of the Φ_u matrix of (4.2). We also define the transition matrix, Φ, as

$$\Phi = I + A \tag{4.16}$$

The nonzero elements of A are given in (4.17).

$$\begin{aligned} &A\ (1,3) = A\ (2,4) = A\ (3,5) = A\ (4,6) = \Delta \\ &A\ (1,5) = A\ (2,6) = \Delta^2/2 \\ &A\ (5,5) = A\ (6,6) = -\Delta/\tau_a \\ &A\ (7,7) = -\Delta/\tau_r \\ &A\ (8,8) = -\Delta/\tau_\psi \\ &A\ (9,9) = A\ (10,10) = -\Delta/\tau_w \end{aligned} \tag{4.17}$$

In (4.17)

Δ = period over which the transition matrix is used.

τ_a = time constant for acceleration bias colored noise (100 sec.)

τ_r = time constant for TACAN range bias colored noise (1000 sec.)

τ_ψ = time constant for TACAN bearing bias colored noise (1000 sec.)

τ_w = time constant for wind error bias colored noise (100 sec.)

Nominal values for the time constants in the program are shown in parentheses.

The nonzero elements of Φ_u are given in (4.18)

$$\Phi_u(3,3) = \Phi_u(4,4) = \sigma_v\Delta_k$$

$$\Phi_u(5,5) = \Phi_u(6,6) = \sigma_a\sqrt{2\Delta_k/\tau_a}$$

$$\Phi_u(7,7) = \sigma_r\sqrt{2\Delta_k/\tau_r}$$

$$\Phi_u(8,8) = \sigma_\psi\sqrt{2\Delta_k/\tau_\psi}$$

$$\Phi_u(9,9) = \Phi_u(10,10) = \sigma_w\sqrt{2\Delta_k/\tau_w} \tag{4.18}$$

In (4.18)

Δ_k = period of the time update (1 second)

σ_v = standard deviation (std) of velocity noise (.0762m/s)

σ_a = std of acceleration bias colored noise (.1524m/s)

σ_r = std of TACAN range bias colored noise (304.8m)

σ_ψ = std of TACAN bearing bias colored noise (2 deg)

σ_w = std of wind bias colored noise (6.1m/s)

Nominal values for the standard deviations are given in parentheses.

4.3 Measurement Models

The measurement models are required for:

1. defining the computed measurement as a function of the estimated state (i.e., $Y(\hat{x},t)$ of (4.7)),
2. defining the partial which relates the residual to the error state (i.e., H of (4.8)) and
3. defining the variance of the random error in measurement (i.e., Q of (4.12)).

The models used in the onboard program are developed in this section for TACAN, MODILS and the airspeed measurements.

TACAN Model

TACAN measurements consist of (a) the range from the aircraft to the station and (b) the bearing (with respect to magnetic north) of the station with respect to the aircraft. The range measurement is assumed to obey

$$Y_{tr} = \sqrt{(x-x_T)^2 + (y-y_T)^2 + (z-z_T)^2} + b_r + q_{tr} \qquad (4.19)$$

where

x,y,z = the coordinates of the aircraft with respect to the runway reference

x_T, y_T, z_T = the coordinates of the TACAN station with respect to the runway reference

b_r = the bias error in the range measurement

q_{tr} = the random error in the range measurement

The estimated measurement is computed from

$$\hat{Y}_{tr} = \sqrt{(\hat{x}-x_T)^2 + (\hat{y}-y_T)^2 + (\hat{z}-z_T)^2} + \hat{b}_r \qquad (4.20)$$

where

$\hat{x}, \hat{y}, \hat{b}_r$ are state variables of the filter and $\hat{z}$ is obtained from the complementary filter (see Reference 2).

The nonzero elements of the row vector, H of (4.8) are calculated from

$$H_{tr}(1) = (\hat{x}-x_T)/(\hat{Y}_{tr} - \hat{b}_r)$$

$$H_{tr}(2) = (\hat{y}-y_T)/(\hat{Y}_{tr} - \hat{b}_r)$$

$$H_{tr}(7) = 1 \qquad (4.23)$$

The variance of the random error in the TACAN range measurement, Q_{tr}, is assumed to be a constant given by

$$Q_{tr} \cong (46m)^2 \qquad (4.24)$$

The bearing measurement is assumed to obey

$$Y_{tb} = \tan^{-1}\left[\frac{(y_T-y)}{(x_T-x)}\right] + \psi_r + b_\psi + q_{tb} \qquad (4.25)$$

where

ψ_r = the azimuth of the runway with respect to magnetic north

b_ψ = the bias error in the bearing measurement

q_{tb} = the random error in the bearing measurement

The estimated measurement is computed from

$$\hat{Y}_{tb} = \tan^{-1}\left[\frac{(y_T - \hat{y})}{(x_T - \hat{x})}\right] + \psi_r + \hat{b}_\psi \tag{4.26}$$

where $\hat{x}$, $\hat{y}$ and $\hat{b}_\psi$ are state variables of the filter.

The nonzero elements of the row vector, H of (4.8) for the bearing measurement are calculated from

$$H_{tb}(1) = (y_T - y)/[(x-x_T)^2 + (y-y_T)^2]$$

$$H_{tb}(2) = (x-x_T)/[(x-x_T)^2 + (y-y_T)^2]$$

$$H_{tb}(8) = 1 \tag{4.27}$$

The variance of the random error in the TACAN bearing measurement, Q_{tb}, is assumed to be a constant given by

$$Q_{tb} = (.5\text{ deg})^2 \tag{4.28}$$

MODILS Model

The MODILS measurements used in the onboard Kalman filter are range and azimuth from a colocated DME transponder and azimuth scanner. The range measurement is assumed to obey

$$Y_{mr} = \sqrt{(x-x_m)^2 + (y-y_m)^2 + (z-z_m)^2} + q_{mr} \tag{4.29}$$

where

x,y,z = coordinates of the aircraft with respect to the runway reference

x_m, y_m, z_m = coordinates of the transponder and scanner with respect to the runway reference

q_{mr} = the random error in the range measurement.

The estimated measurement is computed from

$$\hat{Y}_{mr} = \sqrt{(\hat{x}-x_m)^2 + (\hat{y}-y_m)^2 + (\hat{z}-z_m)^2} \qquad (4.30)$$

where

$\hat{x}, \hat{y}$ are state variables of the filter and $\hat{z}$ is obtained from the complementary filter.

The nonzero elements of the row vector, H, for the range measurement are calculated from

$$H_{mr}(1) = (\hat{x}-x_m)/\hat{Y}_{mr}$$

$$H_{mr}(2) = (\hat{y}-y_m)/\hat{Y}_{mr} \qquad (4.31)$$

The variance of the random error in the range measurement is assumed to be a constant given by

$$Q_{mr} = (18.3m)^2 \qquad (4.32)$$

The MODILS azimuth measurement is assumed to obey

$$Y_{ma} = \tan^{-1}[(y-y_m)/\sqrt{(x-x_m)^2 + (z-z_m)^2}] + q_{ma} \qquad (4.33)$$

where

q_{ma} = the random error in the azimuth measurement

The estimated measurement is computed from

$$\hat{Y}_{ma} = \tan^{-1}[(\hat{y}-y_m)/\sqrt{(\hat{x}-x_m)^2 + (\hat{z}-z_m)^2}] \qquad (4.34)$$

The nonzero elements of the row vector, H, of (4.8) for the azimuth measurement are given by

$$H_{ma}(1) = (\hat{y}-y_m)(\hat{x}-x_m)/(r_1\ (r)^2)$$

$$H_{ma}(2) = r_1/r^2 \tag{4.35}$$

where

$$r = \sqrt{(\hat{x}-x_m)^2 + (\hat{y}-y_m)^2 + (\hat{z}-z_m)^2}$$

$$r_1 = \sqrt{(\hat{x}-x_m)^2 + (\hat{z}-z_m)^2}$$

The variance of the random error in measurement is assumed to be a constant given by

$$Q_{ma} = (.1\ \text{deg})^2 \tag{4.36}$$

True Airspeed Model

As was mentioned in Section III, the existing software calculates the level components of true airspeed in the runway reference from air data and attitude data. These calculated components are assumed to be measurements in the onboard filter rather than using a more complex mechanization involving actual raw data sensors.

The along runway measurement of true airspeed is assumed to obey

$$Y_{ax} = v_x - w_x + q_{ax} \tag{4.37}$$

where

v_x = ground velocity along the runway

w_x = wind velocity along the runway

q_{ax}= the random error in the measurement

The estimated measurement is computed from

$$\hat{Y}_{ax} = \hat{v}_x - \hat{w}_x \tag{4.38}$$

where

$\hat{v}_x$ and $\hat{w}_x$ are state variables of the filter

The nonzero elements of the row vector, H, of (4.8) are given by

$$H(3) = 1 \tag{4.39}$$
$$H(9) = -1$$

The variance of the random error in the measurement is assumed to be a constant given by

$$Q_{ax} = (.61\text{m/s})^2 \tag{4.40}$$

The normal to runway measurement of true airspeed is assumed to obey

$$Y_{ay} = v_y - w_y + q_{ay} \tag{4.41}$$

where

v_y = ground velocity normal to runway

w_y = wind velocity normal to runway

q_{ay} = random error in the measurement

The estimated value of the measurement is computed from

$$\hat{Y}_{ay} = \hat{v}_y - \hat{w}_y \tag{4.42}$$

where

$\hat{v}_y$ and $\hat{w}_y$ are state variables of the filter

The nonzero elements of the row vector, H, of (4.8) are given by

$$H_{ay}(4) = 1$$

$$H_{ay}(10) = -1 \tag{4.42}$$

The variance of the random error in the measurement is assumed to be a constant given by

$$Q_{ay} = (.61m/s)^2 \tag{4.43}$$

4.4 Measurement Preprocessing

The onboard filter contains routines for calculating the residuals and partials (H vector) and summing the results appropriately at a 10Hz frequency. Each residual sum and its partial are transferred to appropriate arrays for processing by the Kalman filter algorithm at a 1Hz frequency.

The preprocessing routines contain the logic for executing the following steps in a sequential manner for each measurement:

1. Test of hardware validity flags. If measurement is invalid, the subsequent steps are bypassed. This step is omitted for the airspeed measurement since there are no hardware validity flags.
2. Calculation of the residual (measurement minus computed measurement).
3. Test of the reasonableness of the residual. If the residual magnitude exceeds a precomputed tolerance level the subsequent steps are bypassed.

4. Accumulation of the residual into the residual sum.

5. Calculation of the H vector for the measurement and referencing the vector to time t_k by $H_m = H(t)\Phi$.

6. Accumulation of elements of H_m into the partial sum.

7. Incrementing a measurement counter by unity. (The number of valid measurements in each sum is calculated).

The TACAN bearing and MODILS azimuth measurements have additional logic before step (2) which rejects the measurement if the ground distance from the station (or scanner) to the aircraft is less than 305 meters.

Following completion of the above logic for all the measurements, a marker is tested to determine if an incremental state change is ready for input to the state. These changes are calculated by the 1Hz filter logic as was described in Section (4.1). If the change is ready then all the residual sums are modified in accordance with linear theory to account for this change (see Equation 4.9). Every second the residual sum and partials are transferred to arrays used in the 1Hz filter logic for calculating the incremental state.

The variance of the random error in each residual sum is calculated from

$$Q^i = Q^i_{xx} \; n^i_x \qquad (4.44)$$

where

Q^i_{xx} = the variance for an individual measurement (see Section 4.3)

n^i_x = the number of residuals in the sum.

4.5 Measurement Rejection

In addition to the validity flags and residual reasonableness tests mentioned in the previous section, a test is made on the reasonableness of the residual sum before it is used to calculate an incremental state change. The Potter algorithm (see Equation (4.14)) requires calculation of the quantity

$$(\sigma_m)^2 = \zeta^T \zeta + Q \qquad (4.45)$$

Let

Δy_m = the residual sum for the particular σ_m involved.

Then the onboard filter rejects the measurement if

$$c|\Delta y_m| > \sigma_m. \qquad (4.46)$$

The best value of the coefficient, c, has not been determined. A value in the range of .25 → .5 is believed to be reasonable.

The value of σ_m for each measurement is stored and used in the reasonableness test prior to summing the residual (Step 3 of Section 4.4).

4.6 Filter Initialization

The onboard mechanization is arranged such that the start of initialization or reinitialization occurs consistent with that of the existing complementary filter. This was set up to give valid comparison of the performance of the two filters during the simulation and flight test phases.

The initialization of the Kalman filter consists of the following:

1. Setting the position (x,y) state variables from TACAN and barometric altimeter data,
2. setting the velocity (v_x, v_y) state variables at the runway referenced true airspeed values,
3. setting wind, acceleration bias and TACAN measurement bias state variables zero, and
4. setting the initial square root matrix in a manner consistent with the above.

The position components are calculated from

$$x = x_t - r_c \cos(\Delta\psi)$$

$$y = y_t - r_c \sin(\Delta\psi) \qquad (4.47)$$

where

$$r_c = \sqrt{(Y_{tr})^2 - (h_b)^2}$$

h_b = altitude above TACAN station computed from barometric altitude

$\Delta\psi = Y_{tb} - \psi_r$

The velocity components are given by

$$v_x = Y_{ax} \qquad (4.48)$$

$$v_y = Y_{ay}$$

The nonzero elements of the initial square root matrix are given by the following:

(1,1)	$= \cos(\Delta\psi)\sigma_{rb}$	σ_{rb} = std of range bias ≃ 305m
(1,2)	$= \sin(\Delta\psi)\sigma_{rb}$	
(1,7)	$= \sigma_{rb}$	
(2,1)	$= \cos(\Delta\psi)\sigma_{rr}$	σ_{rr} = std of random error in the range measurement ≃ 37m
(2,2)	$= \sin\ (\Delta\psi)\sigma_{rr}$	
(3,1)	$= (y_T - y)\sigma_{\psi b}$	$\sigma_{\psi b}$ = std of bearing bias ≃ 2 deg
(3,2)	$= (x_T - x)\sigma_{\psi b}$	
(3,8)	$= \sigma_{\psi b}$	
(4,1)	$= (y_T - y)\sigma_{\psi r}$	$\sigma_{\psi r}$ = std of random error in bearing
(4,2)	$= (x_T - x)\sigma_{\psi r}$	
(5,3)	$= -\cos(\psi_a - \psi_r)\sigma_{vt}$	σ_{vt} = std of random error in airspeed measurement ≅ .61 m/sec
(5,4)	$= -\sin\ (\psi_a - \psi_r)\sigma_{vt}$	
(6,3)	$= v_y\sigma_{\psi a}$	$\sigma_{\psi a}$ = std of error in heading ≅ 2 deg
(6,4)	$= -v_x\sigma_{\psi a}$	
(7,3)	$= \sigma_{xw}$	σ_{xw} = std of x component of wind ≅ 6.1m/sec

(7,9) $= \sigma_{xw}$

(8,4) $= \sigma_{yw}$ σ_{yw} = std of y component of wind $\cong$ 6.1 m/sec.

(8,10) $= \sigma_{yw}$

where

ψ_a = the magnetic heading measurement at time of initialization.

4.7 Memory and Real-Time Requirements

The complementary filter used in the STOLAND system (see Reference [2]) requires approximately 400 words of instructions and constants and approximately two percent (2%) of the available real-time of the 1819A computer. The Kalman filter described in this report requires approximately 2,200 words of instructions and constants and about 22.7% real-time of the 1819A computer. The real-time is distributed between the different functions as tabulated below.

Function	Real-time (%)
Raw data transfers and navigation equation logic (Figures A.2 and A.3 of Appendix)	.40
Measurement preprocessing logic (Figure A.4 of Appendix)	4.74
Kalman filter 1Hz logic (Figure A.5 of Appendix)	17.56
	22.7

As may be seen, the 1Hz logic requires the largest percentage real-time. If we were to execute this logic at .5Hz, then the real-time requirements would change to those tabulated below.

Function	Real-time (%)
Raw data transfers and navigation logic	.37
Measurement preprocessing	4.56
Kalman filter .5Hz logic	8.78
	13.71

The real-time required for the first two functions reduce slightly since the incremental state changes and residual sum transfers occur half as frequently. The real-time required by the third function is reduced by a factor of 2. As is seen, the real-time requirements of the Kalman filter could be reduced by about 40% by using a two-second filter cycle. Some performance degradation of the Kalman filter would be encountered by increasing the cycle time to two seconds. Information relating the performance and filter cycle time could be a useful result of future effort.

V. SIMULATION RESULTS

NASA facilities for the STOLAND include the STOLAND simulator which was developed to provide validation of flight software and hardware (see Reference [2]). This facility was used to checkout the Sperry 1819A computer program as modified to include the Kalman filter logic. Following checkout of the program a few sample flights were run in order to compare the Kalman filter outputs with the complementary filter. Since this was a simulation, the true positions and velocities were available so that the actual error time histories for both filters could be formed. The desired quantities were all calculated and buffered for output to a line printer. Unfortunately, the output frequency was less than .5Hz so the high frequency characteristics of either filter was not recorded. Time did not permit software changes to provide magnetic tape records or strip chart records to give the high frequency characteristics, so the evaluation was restricted to the low frequency characteristics.

The approach trajectory and MODILS and TACAN station locations used in the simulation evaluation are shown on Figure 5.1. The STOLAND simulator and existing 1819A software are such that one may prescribe a reference path for automatic guidance of the simulated aircraft (see reference [2]). For this example the problem was initiated in a manner that gave an early capture (within 20 seconds) onto automatic guidance which was retained for the remainder of the trajectory. The onboard guidance uses the complementary filter for navigation purposes.

During the first 122 seconds of the trajectory shown on Figure 5.1, both filters use the TACAN measurements. At this time they both switch to the MODILS measurements which are used for the remainder of the flight. The error models used for the simulated navaid measurements are described in

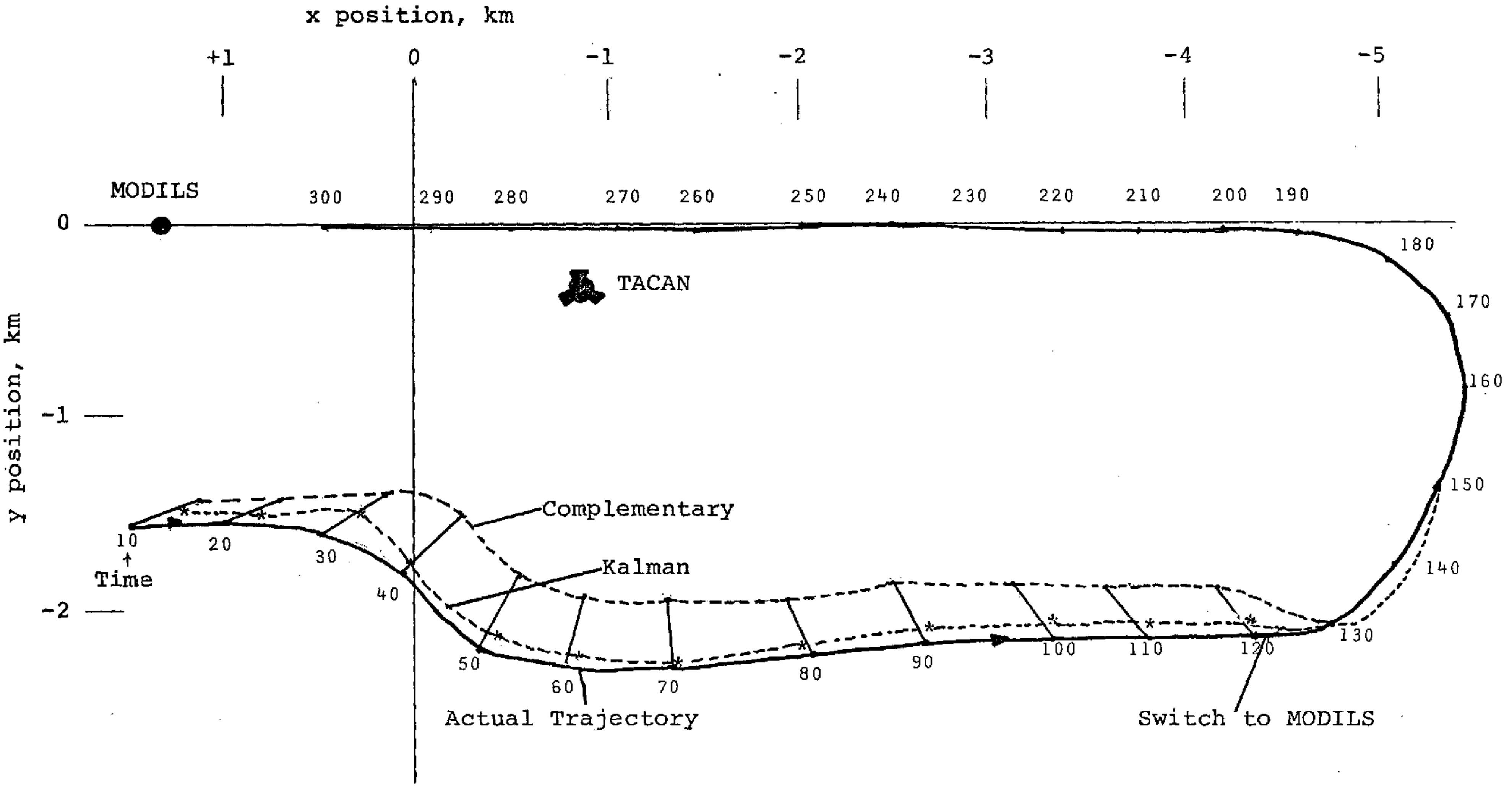


Figure 5.1. Approach Trajectory Used for Filter Evaluation

reference (2). For the specific examples presented here, the TACAN range bias was -305 meters and the TACAN bearing bias was -2°. These biases are the primary contributors to the relatively large errors between the complementary filter outputs and actual trajectory shown on Figure 5.1. The Kalman filter, which includes the bias errors as state variables, has considerably smaller errors.

The velocity time histories for the approach trajectory of Figure 5.1 are shown in Figure 5.2. The complementary filter velocity estimate is seen to deviate considerably from the actual trajectory values. Figures 5.3 and 5.4 give time histories of the position and velocity errors for both filters. The data here was plotted at 10-second intervals and straight lines drawn between points. The error time histories generally illustrate the superior performance of the Kalman filter during TACAN data and for the transition phase in switching between navigation aids. After about 200 seconds the errors for either filter are very small.

Figure 5.5 presents the time histories of the parameter estimates obtained from the Kalman filter. The Kalman filter obtains a reasonably good estimate of the TACAN range bias while using the TACAN data. This is largely a result of the trajectory used which passes by the TACAN station as seen on Figure 5.1. The TACAN bearing bias is partially estimated during the TACAN data phase. The wind estimates shown on Figure 5.5 are actually errors since the winds used in the simulation are zero for this case. The wind errors during the TACAN data phase are primarily due to the ground velocity errors. After switching to MODILS, the errors are smaller and are believed to be caused by software approximations in transforming the airspeed to the runway reference. The acceleration bias time histories shown on Figure 5.5 are also errors since there were no intentional sources of bias in the

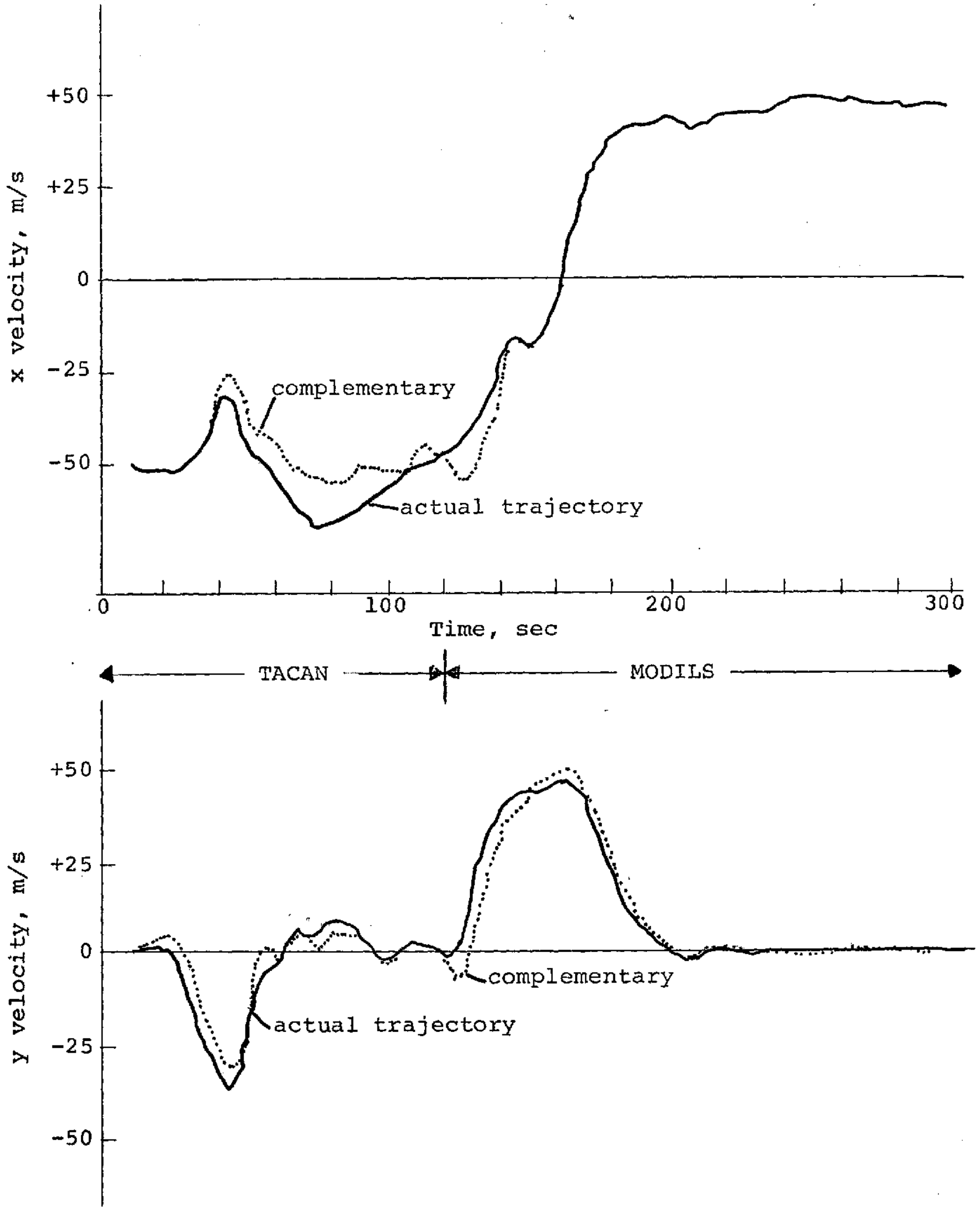


Figure 5.2. Velocity Time Histories for Approach Trajectory

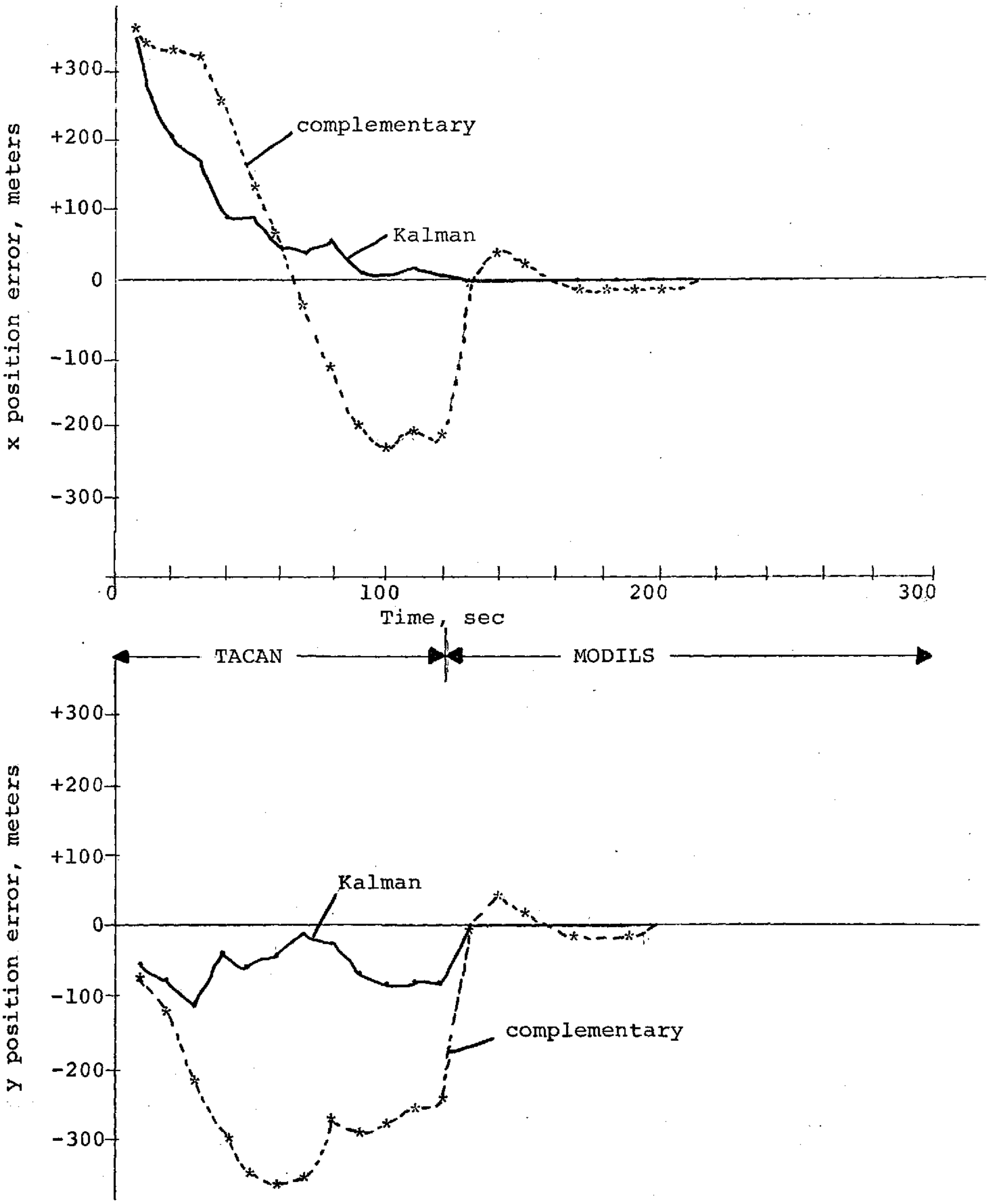


Figure 5.3. Position Errors on the Approach Trajectory

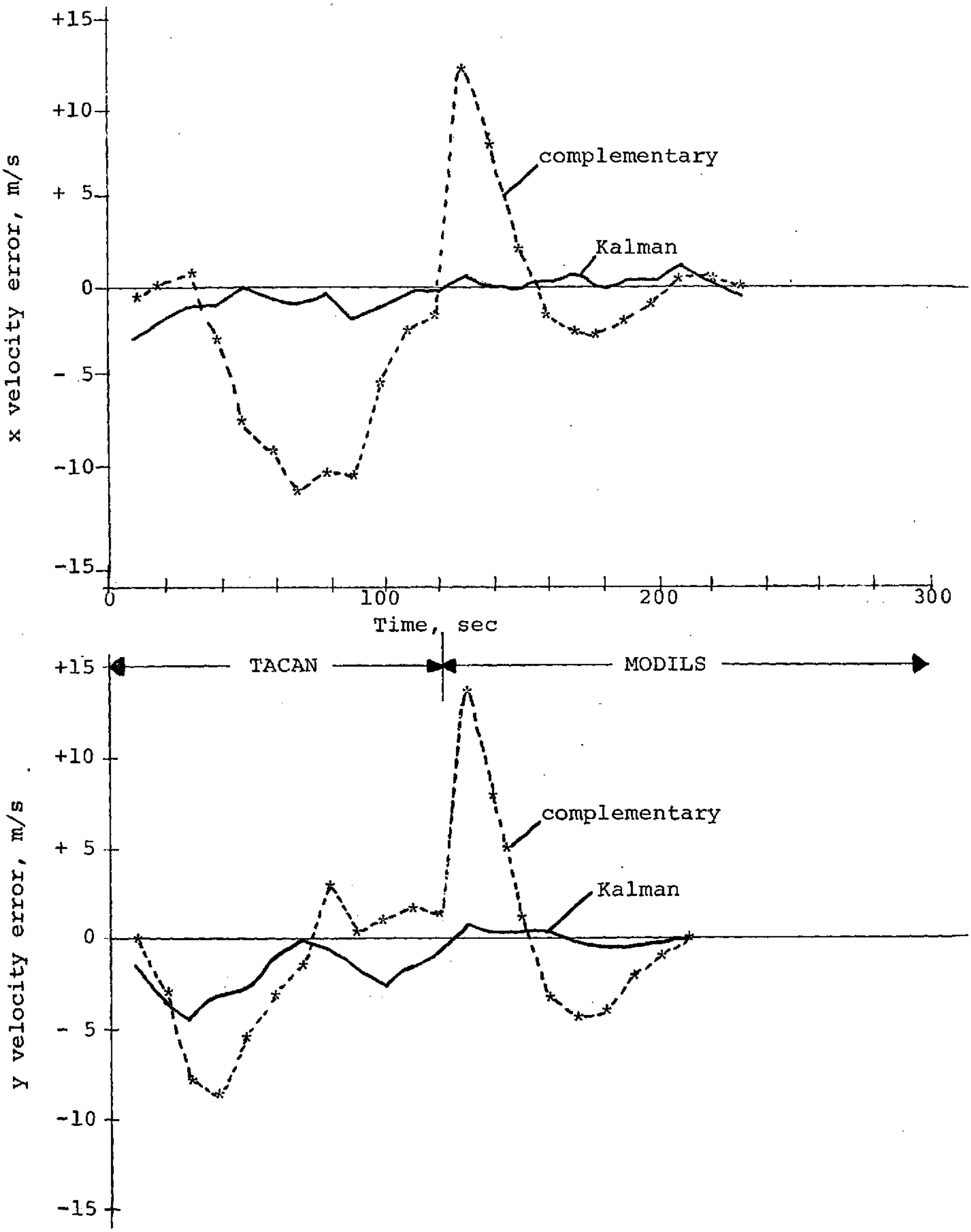


Figure 5.4. Velocity Errors on the Approach Trajectory

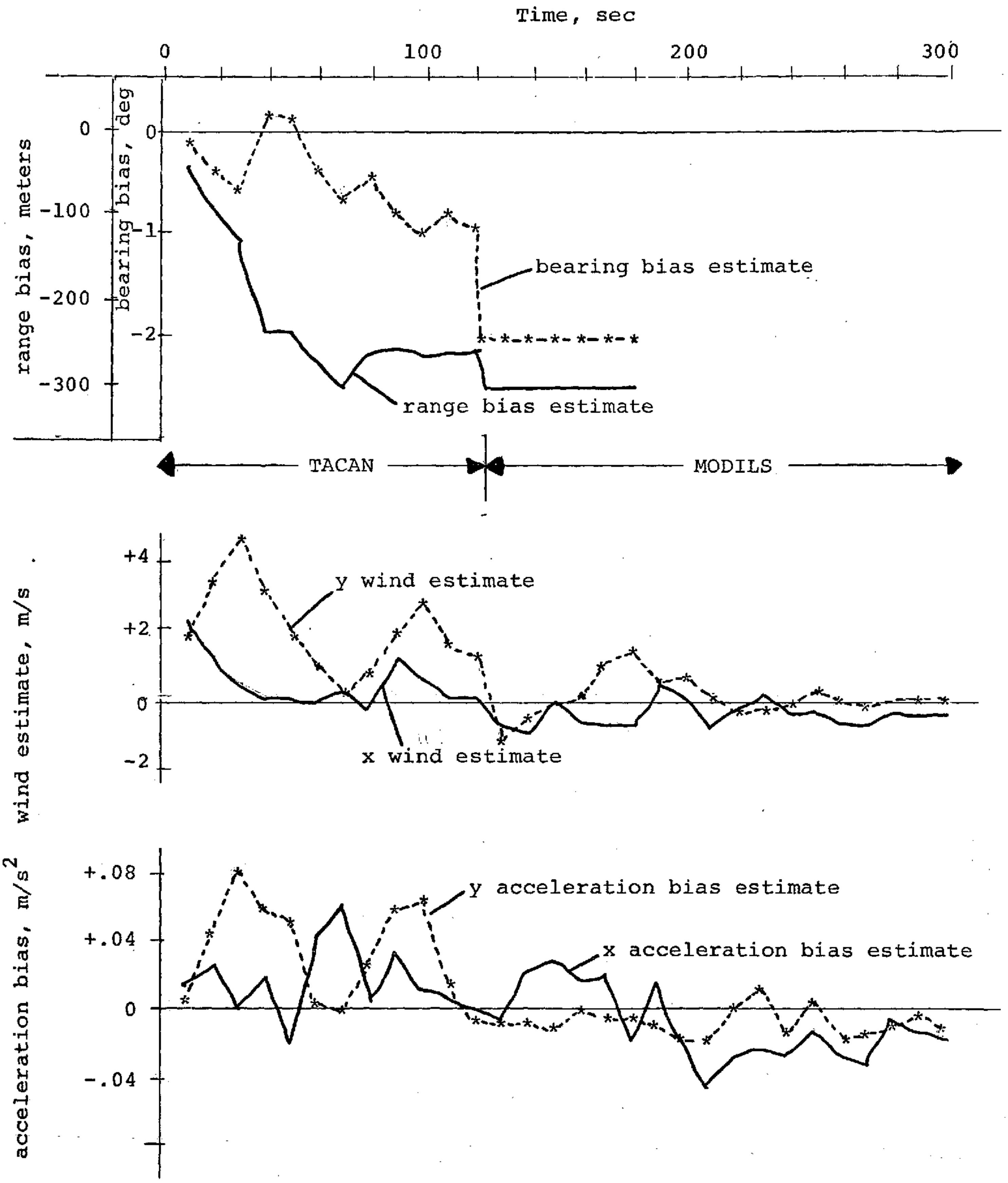


Figure 5.5. Parameter Estimates on the Approach Trajectory

simulation. The estimates obtained from the filter are small and considered acceptable for this example. These states were introduced to compensate for errors in the attitude and heading reference used in transforming the body-mounted accelerometer data to runway reference. How well they compensate for such errors, particularly during maneuvering, needs further investigation.

Figure 5.6 presents the average residuals for the approach trajectory. An examination of the data indicates a standard deviation of about 61 meters for the TACAN range noise and about .08° for the bearing noise. The individual range measurement noise (if random errors are independent) would be about 193 meters rather than the 46 meters indicated in equation (4.24). The TACAN range variance used in obtaining the data shown in Figures 5.1 to 5.5 was obtained with $Q_{tr} = (183m)^2$ rather than $(46m)^2$. The TACAN range variance of $Q_{tr} = (46m)^2$ was obtained from analysis of flight data. The TACAN bearing noise is believed to be too small in comparison to flight experience. These factors indicate the models given in reference [2] should be improved to give a more consistent agreement with the actual flight experience.

Figure 5.7 compares the velocity errors for the two filters on a different run where the Kalman filter measurement variances are as given in section 4.3. The high frequency components of the velocity errors for the Kalman filter are worse than those of the complementary filter. This data led to increasing the TACAN range variance for the run presented in the previous figures.

Figure 5.8 compares the velocity errors in the presence of a 10-knot crosswind and wind gusts. The velocity errors for the Kalman filter after about 220 seconds were slightly higher than those of the complementary filter. This

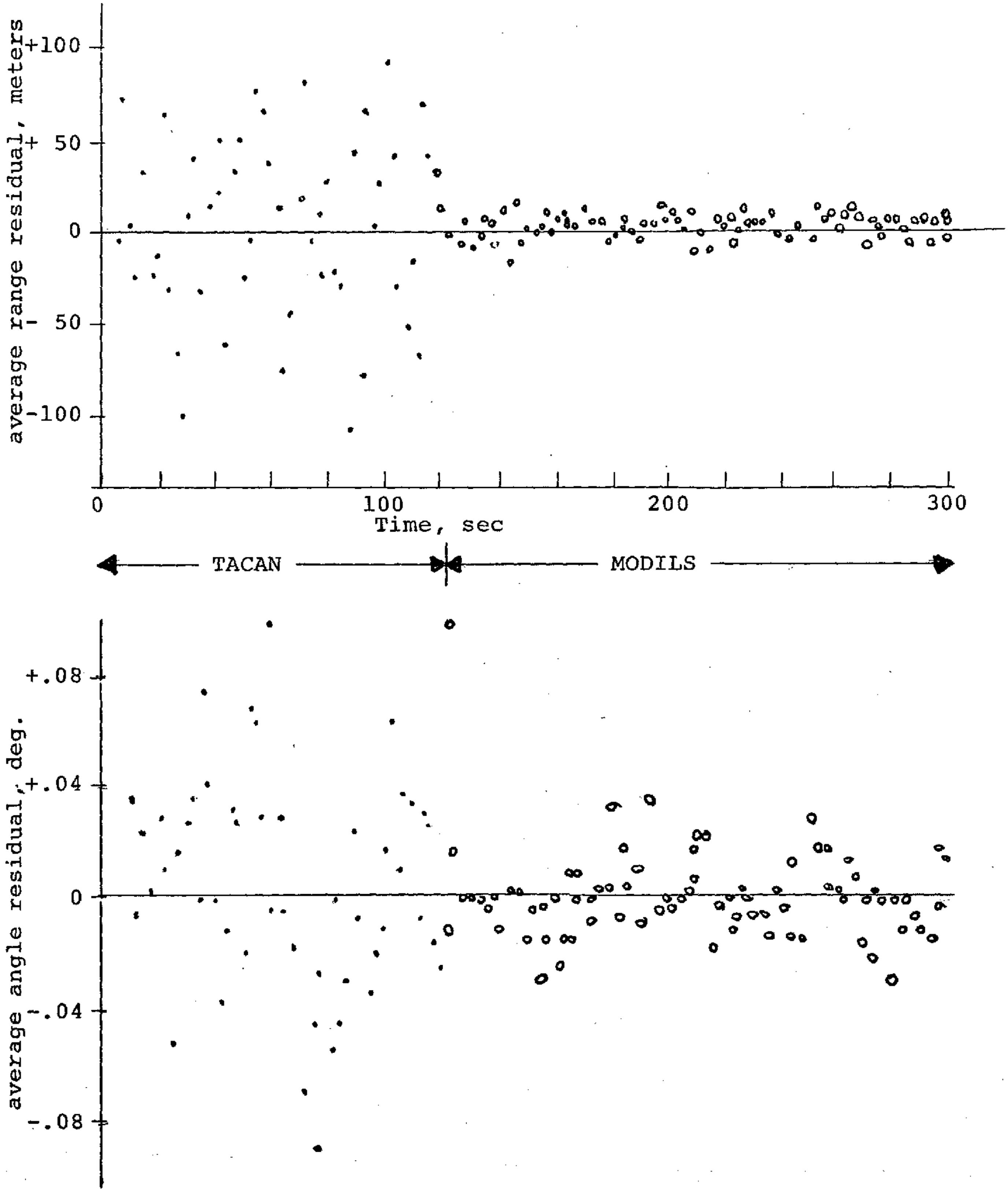


Figure 5.6. Residuals on the Approach Trajectory

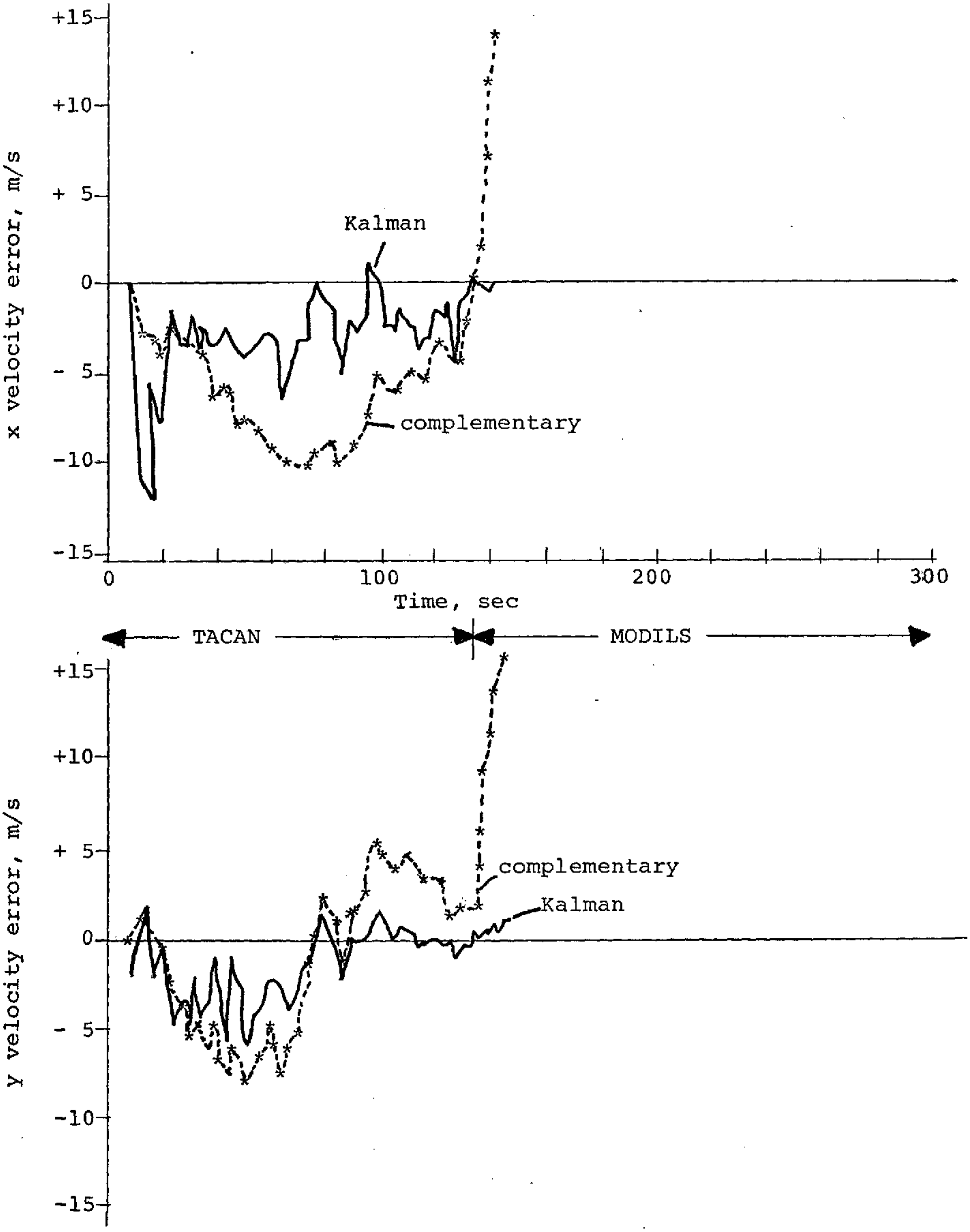


Figure 5.7. Velocity Errors with TACAN Range Variance Too Small

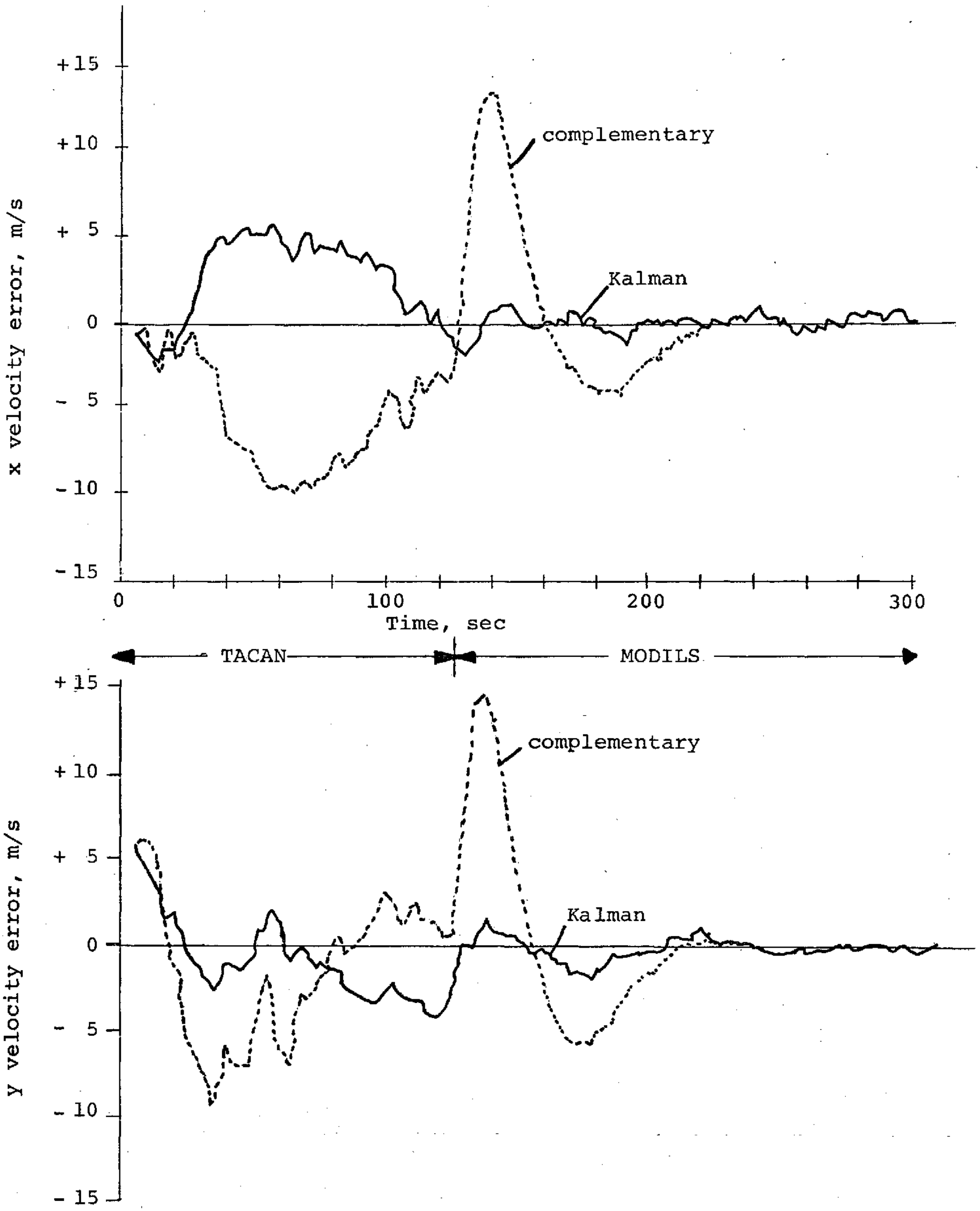


Figure 5.8. Velocity Errors with Cross Wind and Gusts

is believed to be caused by either too high a weight on the true airspeed data or an insufficient compensation for wind changes in the wind model. These factors also need further investigation.

VI. CONCLUDING REMARKS AND RECOMMENDATIONS

At the initiation of the filter development described herein, it was generally known that a Kalman filter could be developed to give improved performance over the complementary filter. The performance improvements are gained, however, at the expense of computer memory and real-time. Whether the Kalman filter could be made simple enough to fit within the available memory and real-time in the STOLAND system computer and still give improved performance over the complementary filter was not known. One of the encouraging results is that the Sperry 1819A computer in STOLAND is sufficiently large and fast to allow a meaningful test evaluation of advanced navigation systems while still performing all the other necessary guidance, display and control calculations of the STOLAND design.

The memory and real-time constraints were satisfied by (1) keeping the number of state variables to a bare minimum, and (2) using data averaging techniques to reduce the number of measurements requiring filter calculations. This design, therefore, uses the accelerometer data for keeping the position and velocity estimates current at the high rate needed for guidance, control and display purposes. Previous designs had used this same philosophy; however, the acceleration data was from high quality inertial components. The inertial reference in STOLAND is a strapped-down type employing relatively inexpensive components whose errors are probably two orders of magnitude greater than inertial quality components. These errors were adequately compensated by the simple filter design with no apparent problems with the one-second delay of the filter calculations. Very likely the strapped-down reference is good enough to use a larger delay and reduce the filter real-time requirements further with only minor performance degradation.

Information relating performance degradation and real-time reductions could be a valuable output from further work.

Some additional areas where further work is recommended are as follows:

1. The existing filter has no means of providing direct (instantaneous) compensation for heading errors in the inertial reference. The addition of this variable to the state vector could give improved performance during maneuvering where heading errors can be moderately large.

2. The airspeed data software should be thoroughly reviewed. It is believed that the wind and velocity estimates could be improved with relatively minor software improvements.

3. The problem of interfacing the Kalman filter with real-time guidance, display and control needs investigation. The current filter should be connected to these functions and any problems with the discrete changes in state of the filter resolved by simulation tests.

4. Potential performance improvements and complexity in adding a vertical channel (z axis) should be studied.

5. A three-axes navigation system employing

a Kalman filter should be designed and interfaced with the STOLAND guidance, control and displays. Flight and simulation tests of such a design would be a good final validation of the advantages/disadvantages associated with a moderately sophisticated navigation system in STOL aircraft.

REFERENCES

[1] Smith, D.W., Neuman, F., Watson, D.M., and Hardy, G. H., "A Flight Investigation of a Terminal Area Navigation and Guidance Concept for STOL Aircraft," NASA TMX-62, 375, July 1974.

[2] Newman, Frank and Warner, David N., Jr., "A STOL Terminal Area Navigation System," NASA TMX-62, 348, May 1974.

[3] Kaminski, P. G., Bryson, A. E., Jr., and Schmidt, S. F., "Discrete Square Root Filtering. A Survey of Current Techniques," IEEE Transactions on Automatic Control, December 1971.

[4] Kaminski, P. G., "Square Root Filtering and Smoothing for Discrete Processes," SUDAAR #427, Stanford University Guidance and Control Laboratory, July 1971.

APPENDIX

Description of the Onboard Program

The Kalman filter described in the body of this report was designed to operate as an experiment within the leftover memory and real-time of the STOLAND Twin Otter software. The experimental objectives were to calculate and record the Kalman filter outputs using the same raw data sources as those of the complementary filter. The complementary filter outputs are also recorded at the same time as those of the Kalman filter; therefore, post-flight evaluation of the flight data will permit direct comparisons of both filter outputs with ground tracking data.

This Appendix describes the Kalman filter logic of the overall program and the executive which provides the time sharing between the existing software and the Kalman filter. A comparison of memory and real-time requirements of the two filters and some details on how the real-time is distributed between different parts of the Kalman filter is also presented.

A.1 Executive Driver

In order to provide the available real-time to the Kalman filter logic, it was necessary to develop a new executive for the Twin Otter program. A macro flow chart of this executive is presented in Figure A.1. At program start initialization logic is executed. The interrupts are then enabled

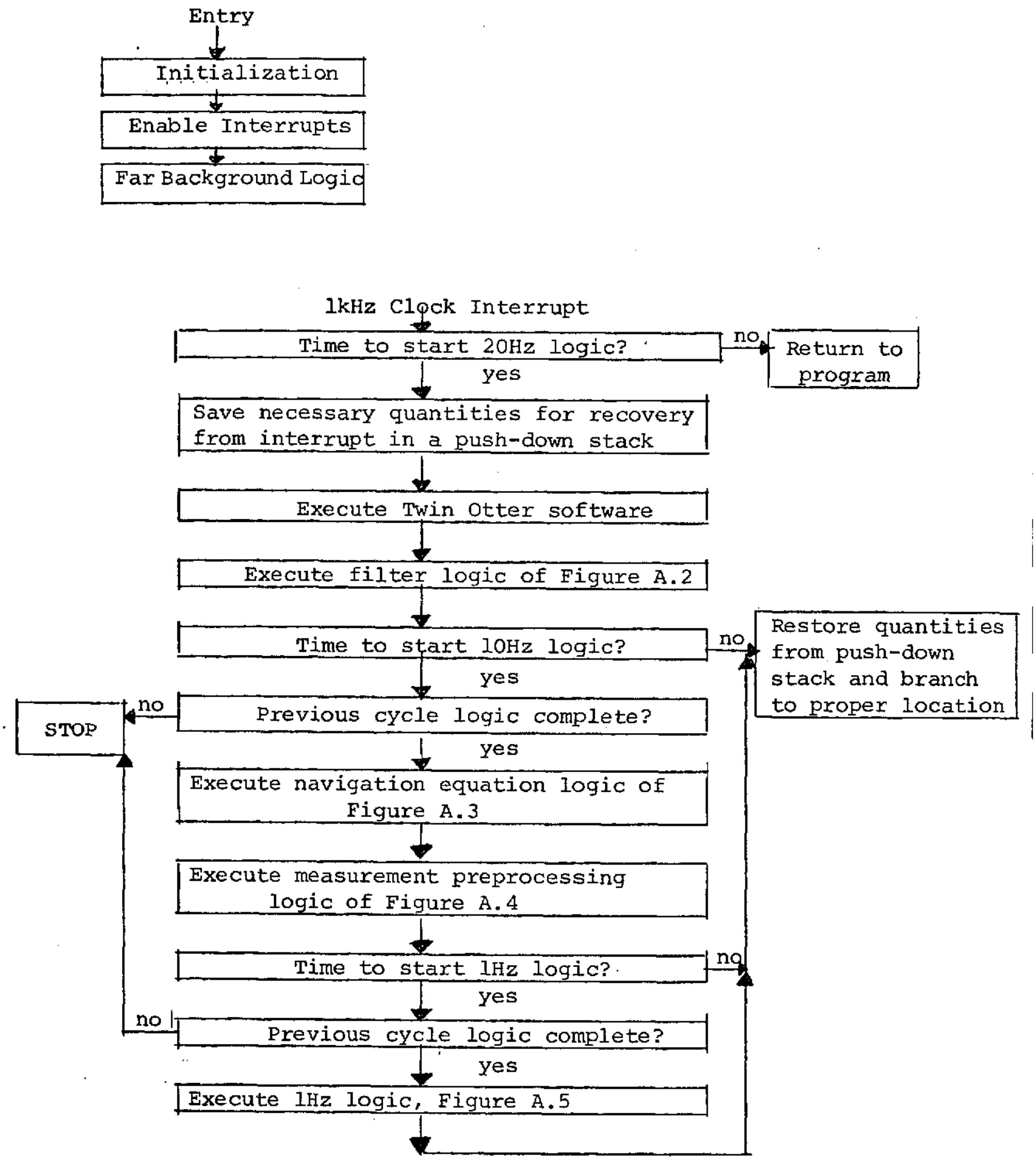

Entry
Initialization
Enable Interrupts
Far Background Logic
1kHz Clock Interrupt
Time to start 20Hz logic?
no
Return to program
yes
Save necessary quantities for recovery from interrupt in a push-down stack
Execute Twin Otter software
Execute filter logic of Figure A.2
Time to start 10Hz logic?
no
Restore quantities from push-down stack and branch to proper location
yes
STOP
no
Previous cycle logic complete?
yes
Execute navigation equation logic of Figure A.3
Execute measurement preprocessing logic of Figure A.4
Time to start 1Hz logic?
no
yes
no
Previous cycle logic complete?
yes
Execute 1Hz logic, Figure A.5

Figure A.1. Macro Flow Chart of Executive for STOLAND Software

and the far background logic is entered. The program waits in the far background routine until an interrupt occurs. The STOLAND system contains several sources of program interrupt; however, we will only describe the main stream of calculations which are triggered by the 1kHz internal clock of the 1819A computer. A clock interrupt causes transfer to a location where a counter is decremented and tested to determine if it is time to initiate the 20Hz calculations. If not, the program returns to the location where interrupted and continues execution.

If it is time to start the 20Hz logic, then the necessary quantities for recovery from the interrupt (registers, location at intercept, etc.) are saved in a software push-down stack. The program then executes the Twin Otter navigation, guidance, control and display software. Following completion, the program executes the software for interfacing the Kalman filter with the overall program. The next logic is a test to determine if it is time (every other cycle) to start the 10Hz calculations of the Kalman filter. If not, the executive restores all the quantities saved in the push-down stack and branches to the saved interrupt location. In this instance, the program branches to the lower priority logic in the order (1) 10Hz (if not complete); (2) 1Hz (if not complete); or (3) far background.

If the 10Hz logic is to be initiated a marker is tested to see if the previous cycle calculations were completed before the interrupt. If not, a STOP is executed since either a software or hardware problem has prevented completion of the 10Hz logic within the allowed time. With no malfunction the program then executes the 10Hz Kalman filter logic. Following completion of the 10Hz logic, a test is made to determine if the 1Hz Kalman filter logic should be initiated. If not, the executive restores all the quantities saved in the push-down stack and branches to the saved interrupt location. In this instance, the program branches to the lowest priority logic in the order (1) 1Hz (if not complete); (2) far background. If the 1Hz logic should be initiated, a marker is tested to see whether the previous cycle calculations were completed in the allowed time. If not, a STOP is executed. If no malfunction, the 1Hz logic is initiated.

Following completion of the 1Hz logic, the executive restores the quantities from the push-down stack and branches to the saved location. In this instance the saved location is in the far background logic.

A.2 20Hz Kalman Filter Logic

The Kalman filter logic which is executed at 20Hz is shown in Figure A.2. The first test is to determine if

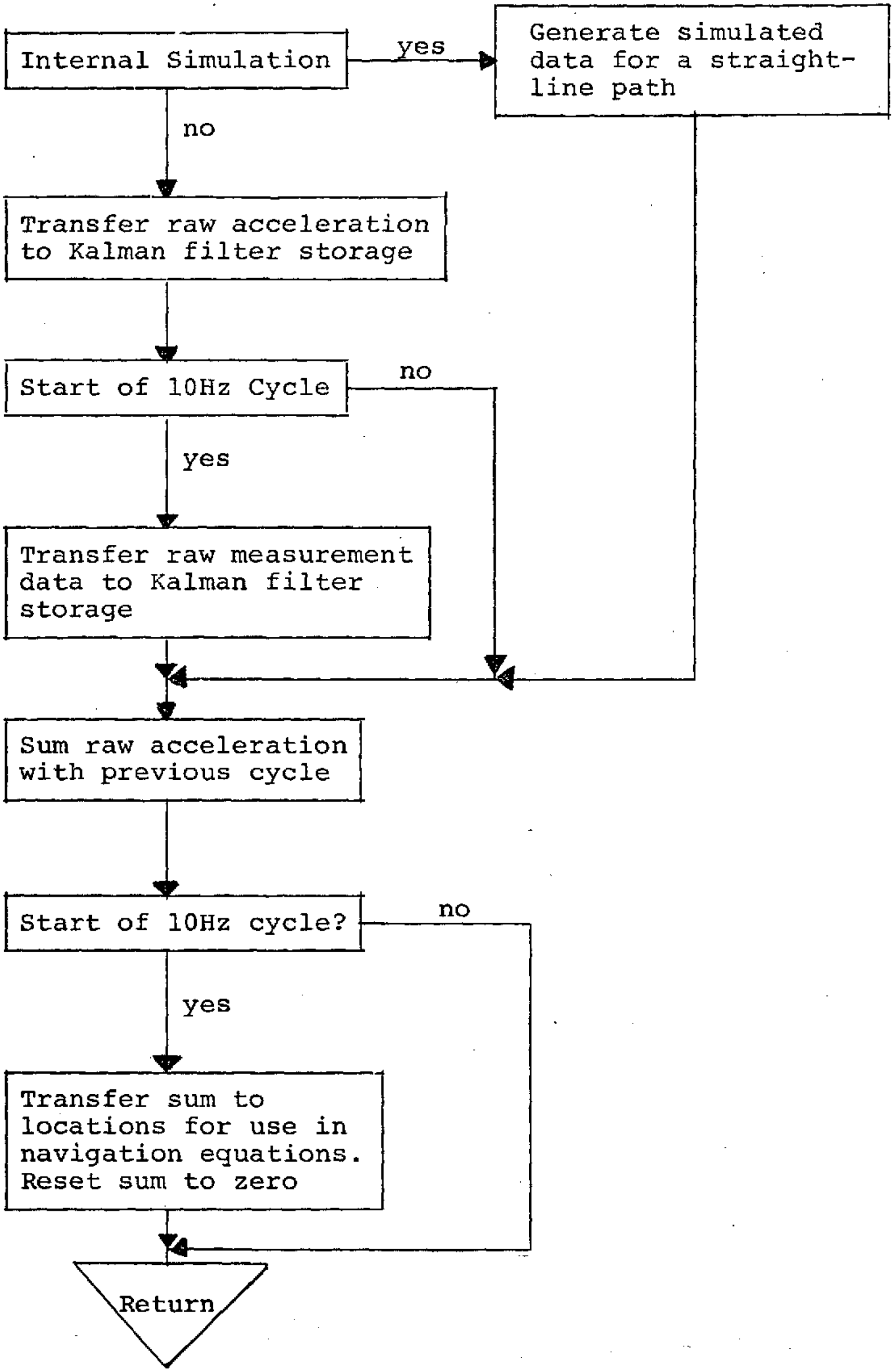

Internal Simulation
yes
Generate simulated data for a straight-line path
no
Transfer raw acceleration to Kalman filter storage
Start of 10Hz Cycle
no
yes
Transfer raw measurement data to Kalman filter storage
Sum raw acceleration with previous cycle
Start of 10Hz cycle?
no
yes
Transfer sum to locations for use in navigation equations. Reset sum to zero
Return

Figure A.2. Macro Flow Chart of Kalman Filter 20Hz Logic

data should be generated internally for a straight-line path. This logic was used for check-out phases and is a convenient way of checking the Kalman filter operations for new assemblies. It can be removed at any time. For normal operation the routine transfers the raw acceleration data from the complementary filter locations to the Kalman filter storage at a 20Hz rate. A test is then made to determine if the particular entry is at the starting time for a 10Hz cycle. If true, then the navigation aid measurements and their validity flags are transferred from locations used by the complementary filter to locations used by the Kalman filter. The raw acceleration data are then accumulated (at 20Hz) with those of the previous cycle and a test made to determine if the entry corresponds to that for starting the 10Hz logic of the filter. If true, the accumulated sum is transferred into locations for use by the navigation equations and the sum locations reset to zero.

A.3 Navigation Equation Logic

Figure A.3 shows the navigation equation logic which is executed at 10Hz. The first operation is a test for determining if initialization is required. This marker is set true when the complementary filter begins its initialization.

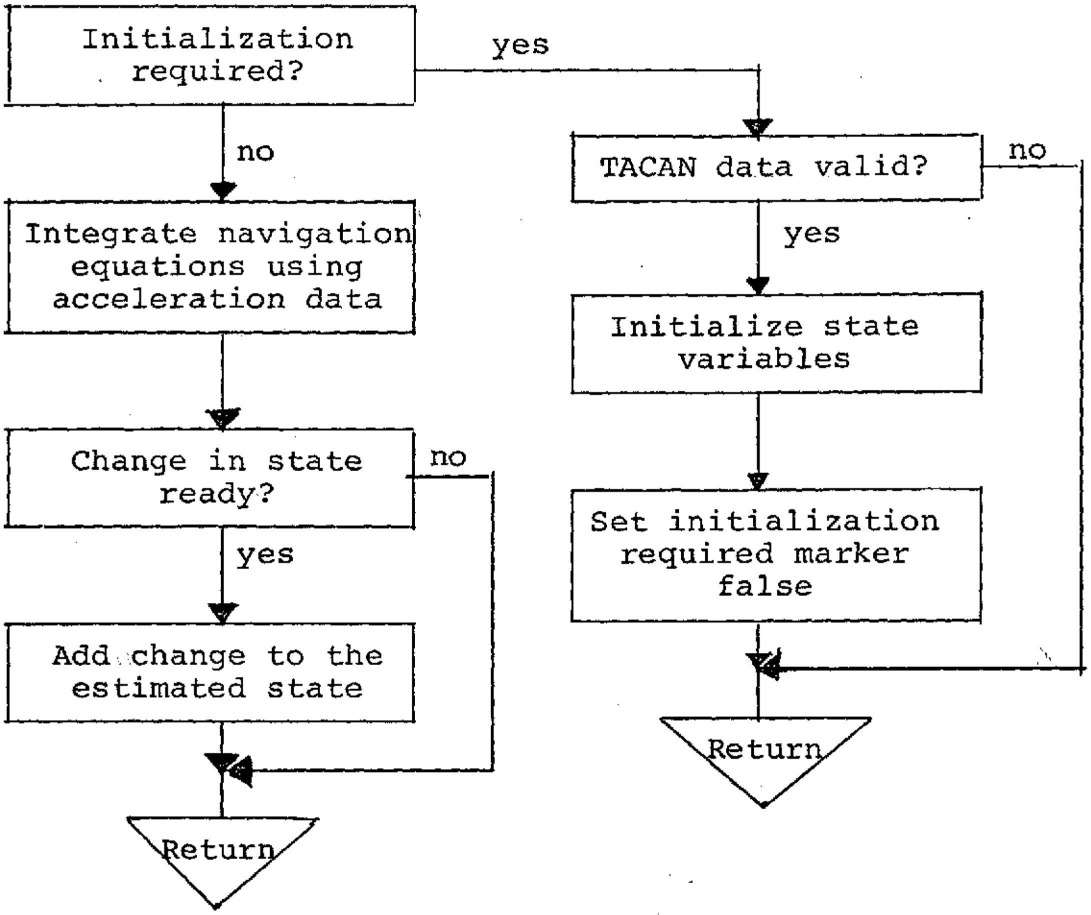

Initialization required?
yes
no
TACAN data valid?
no
yes
Integrate navigation equations using acceleration data
Initialize state variables
Change in state ready?
no
yes
Set initialization required marker false
Add change to the estimated state
Return
Return

Figure A.3. Macro Flow Chart of Navigation Equation Logic.

If initialization is required, then the validity flags for the TACAN data are tested. If both the range and bearing are valid, the state variables are initialized and the initialization marker set false. If TACAN is not valid, the routine returns.

If initialization is not required, the logic integrates for the position and velocity using the acceleration data. Following the integration, a marker is tested to determine if a state change is ready. If it is ready, then the change is added to the estimated state and the routine returns.

A.4 Measurement Preprocessing Logic

Figure A.4 shows the measurement preprocessing logic which is executed at 10Hz. A marker is tested first to see if the preprocessing logic has been initialized. This marker is set true after the state variables have been initialized for the navigation equations (Figure A.3). After initialization (if it is required), the logic does the necessary preprocessing for the TACAN, MODILS, and airspeed data as described in section IV. The transition matrix elements are then updated so the matrix is valid for the next entry. The incremental state change marker is then tested. If true, the residual sums are corrected, and the incremental state change is updated with the transition matrix. The change ready marker is then set true for use by the logic of Figure A.3. Following this, the residual sum cycle marker is tested. Every tenth entry, the marker is true and the logic shown is executed.

A.5 Kalman Filter 1Hz Logic

Figure A.5 shows the Kalman filter logic which is executed at 1Hz. The first test determines if initialization is required. If true, the quantities shown are initialized and the initialize marker set false before return. If the initialize marker is false, a test is made to determine if there are any measurements (residual sums) to be processed. If true, Potter's algorithm (see section 4.1) is used to process each available residual sum in a sequential manner. After completion, the incremental state change marker is set true for use in the logic of Figure A.4. Following this, the square root matrix is updated to the beginning of the next measurement accumulation period (see section 4.1).

A.6 Summary of Kalman Filter Routines

Table A.1 presents the name and function of the Kalman filter routines developed for the Sperry 1819A computer.

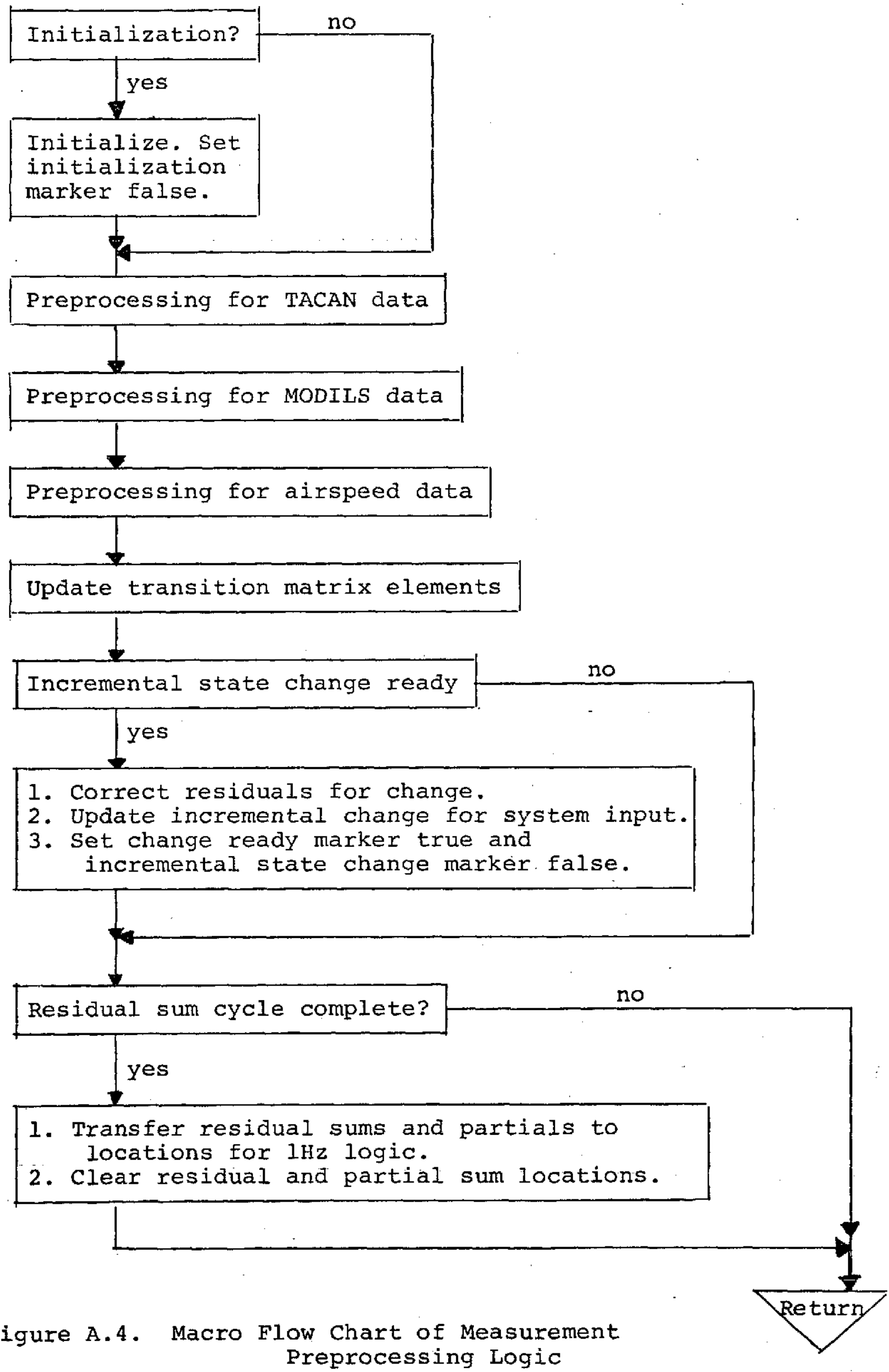

Initialization?
no
yes
Initialize. Set initialization marker false.
Preprocessing for TACAN data
Preprocessing for MODILS data
Preprocessing for airspeed data
Update transition matrix elements
Incremental state change ready
no
yes
1. Correct residuals for change.
2. Update incremental change for system input.
3. Set change ready marker true and incremental state change marker false.
Residual sum cycle complete?
no
yes
1. Transfer residual sums and partials to locations for 1Hz logic.
2. Clear residual and partial sum locations.
Return

Figure A.4. Macro Flow Chart of Measurement Preprocessing Logic

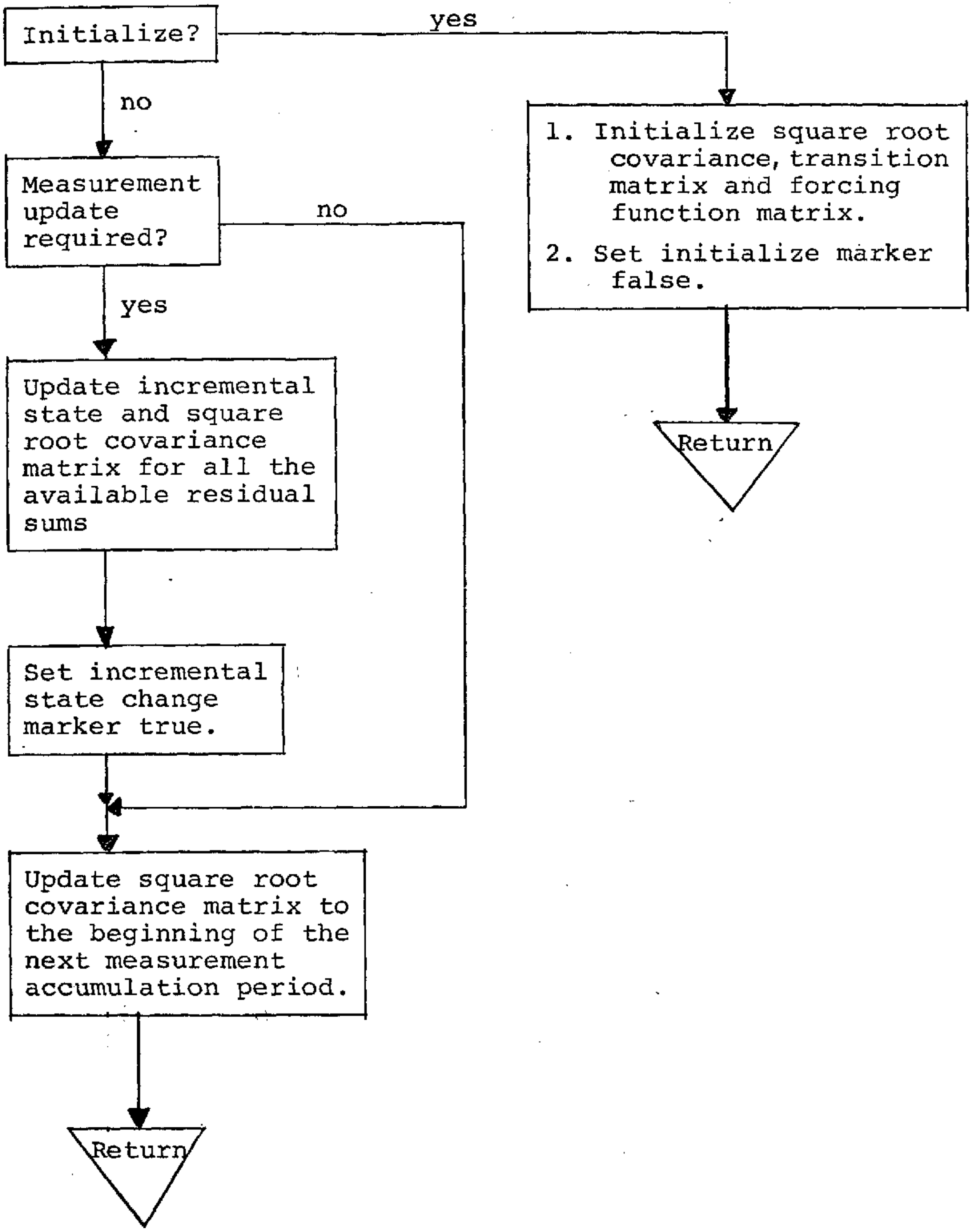

Initialize?
yes
no
1. Initialize square root covariance, transition matrix and forcing function matrix.
2. Set initialize marker false.
Return
Measurement update required?
no
yes
Update incremental state and square root covariance matrix for all the available residual sums
Set incremental state change marker true.
Update square root covariance matrix to the beginning of the next measurement accumulation period.
Return

Figure A.5. Macro Flow Chart of the Kalman Filter 1Hz Logic.

Twenty-one (21) routines are involved in the Kalman filter logic. The 1819A assemblies of these routines contain a number of descriptive comments for clarifying the function of the 1819A instructions.

TABLE A.1. Kalman Filter Routines

Name	Function
AIRRES	Computes airspeed residuals and partials and accumulates results into preprocessing storage locations.
CRESID	Corrects all residual sums for the incremental state change.
DATTRF	Transfers raw data from locations of the complementary filter to locations used by the Kalman filter (see Figure A.2).
FILNIT	Initializes the square root covariance matrix, the transition matrix and the matrix of sensitivity to forcing functions.
HTPHI	References measurement partials to the filter reference time and accumulates result with the previous sum.
KALM	This is the driver for the 1Hz logic of Figure A.5.
LOADR	Transfers residual sums and their partials to arrays used by the 1Hz Kalman filter logic. Clears accumulation locations used by the measurement preprocessing logic.
MODRES	Performs all preprocessing calculations for MODILS range and azimuth measurements.
MCWT	Executes Potter's algorithm for calculating the incremental state and the square root covariance for each measurement (residual sum).
MESCAL	This is the driver for the measurement preprocessing logic of Figure A.4.

NAVEQ This routine executes the navigation equation logic of Figure A.3.

PHIX Updates the incremental state change over one filter cycle (1 second).

REDWT Reduces square root covariance to upper triangular form using Householder's Algorithm.

RND, RNDA Rounds A register into single precision in AU.

SETUP Initialization for filter markers, printer and straight-line flight simulation.

SIMMES Computes simulated measurements for a straight-line flight.

SUMACC Accumulates acceleration data at 20Hz and transfers values to filter at 10Hz.

TACRES Performs all preprocessing calculations for TACAN range and bearing measurements.

TIMUPD Updates transition matrix elements at 10Hz.

UPDATX Updates the incremental state change for a fraction of the filter cycle.

UPTWT Forms the updated square root covariance in expanded form.

A.7 Summary of Equations

The equations solved in the 1819A computer are summarized below.

<u>20Hz</u> Accumulate measured accelerations

$a_s^1 = a_s^1 + \ddot{x}_r$ x axis acceleration sum

$a_s^2 = a_s^2 + \ddot{y}_r$ y axis acceleration sum

<u>10Hz</u> ($\Delta =$ .1 sec.) Update state equations

$x^i(t+\Delta) = x^i(t) + [a_s^{i-2}/2 + x^{i+2}(t)]\Delta$ update velocities $i=3,4$

$x^i(t+\Delta) = x^i(t) + [x^{i+2}(t+\Delta) + x^{i+2}(t)]\Delta/2$ update positions $i=1,2$

$x^i(t+\Delta) = x^i(t)$ update states $i=5,10$

Measurement preprocessing; Executed for all valid measurements (k=1,6)

$\Delta y^k = Y^k - \hat{Y}^k$ compute residual

$y_s^k = y_s^k + \Delta y^k$ residual sum

$H^k = \nabla_x \hat{Y}^k$ compute gradient (partial)

$H_m^k = H_m^k \Phi$ refer partial to beginning of 1Hz cycle

$H_s^k = H_s^k + H_m^k$ partial sum

<u>1Hz</u> (Δ= 1 sec.) Update incremental state and covariance matrix using Potter's algorithm: Executed for each of the four measurements in effect (MODILS + airdata or TACAN + airdata)

$S_k = H_s^k W W^T (H_s^k)^T + Q^k$ variance in residual

$d\hat{x} = d\hat{x} + W W^T (H_s^k)^T [y_s^k - H_s^k d\hat{x}]/S_k$ update incremental state vector

$W^T = W^T - W^T (H_s^k)^T H_s^k W W^T / [S_k(1 + \sqrt{Q^k/S_k})]$ update square root covariance matrix

Add incremental state to state estimate (i=1,10)

$d\hat{x}(t) = \Phi d\hat{x}$ refer state change to current time

$x^i = x^i + d\hat{x}^i$ update states

$d\hat{x}^i = 0$

Update W^T with forcing functions

$$W^T(t+\Delta) = \begin{bmatrix} W^T(t)\Phi \\ \Phi_u \end{bmatrix}$$

$W^T(t+\Delta)$ → upper triangular form using Householder's algorithm.